



PC-8801mkIISR

やさしいマシン語



はじめに

1885年1月に発表されたPC-8801MkIJSRは、PC88シリーズの3代目にあたります。SRは高鮮像度高速多機能グラフィックスの完備やFM音楽ユニットの 形板とどの性能を持つ。8ビットとしては完成度の高いパソコンです。これらの優れた性能を100 5引き出すには、BASICだけでは不可能です(せいせい60 %といったところでしょうか)。それは、BASICではハードウェアに関する細かいプログラミングができないからです。もし、BASICで細かいプログラミングをしたいのであれば、BASICコンドの中にあるマシン語を操作するコマンド (USR、POKE、PEEKなど)を使用しなければなりません。当然、これらのBASICコマンドを使うためにはマシン語の製造が必要となります。

またBASICはインタプリタなので実行速度が遅いという欠点を持っています。 実行速度の速いマシン語を使うと、この欠点もなくなります。

本書はマシン語をやさしく、かつ、より深く理解できるように執筆したもの で、標成もこれに沿っています。前年はマシン語の基礎から入ってだんだんと 理解を深めるようにしてあります。そして、後半の「マシン語アラカルト」で 南半で説明できなかったことや、筆者の勤務先の科学部の生徒達が日頃疑問に 思っていることについて説明しています。

マシン語を早く覚えるコツは、本書のプログラムを理解したら一部を変えたり、同じ動きをするプログラムを別に作ったりしてゆくことです。 読者の皆さんが早くマシン語を理解し、SRの性能を100%引き出せることを下された事態。

最後に、執筆のお勧めを頂き、遅筆の筆者を絶えず励まして下さった技術評 論社の編集スタッフ諸氏に感謝申し上げる次第です。

1985年 7月 前田 光男



CONTENTS

第1章 初級者から中級者までのマシン語 ~誰もが知っておきたい基礎知識

1	2進数と16進数15
	2 進数は0と1の世界15
	ピットとバイト16
	16進数を使って 2 進数を読みやすく17
2	Z-80A CPU20
	Z-80Aは8ピットCPUの主流 ······20
	Z-80Aの内部構成・・・・・・20
	マシン語命令の長さ23
	Fレジスタを探る25
3	PC-8801mkIIの
	ハードウェアを知ろう28
	メモリはどう使われているか28
	システムからVRAMまでコントロールする
	1/0アドレスマップ29
	N ₈₈ -DISK BASIC使用時のメモリマップ30
4	マシン語モニタを
	使いこなすために33
	非常に強力なマシン語モニタ33
	いよいよマシン語の世界へ35
	マシン語モニタ36
	各コマンドはこう使う39
5	アセンブリ言語を使うために59

	CPUによるアセンブリ言語の違いと共通点 アセンブリ言語表現の約束事	
	マシンにない擬似命令	
	アセンブリ言語のかたち	64
	ハンド・アセンブルでマシン語マスター…	
第2章	モノクロ画面からカラー画面 ~キャラクタ単位	
1	CRTのカラーの原理······	69
	CRTの基本は3原色	69
2	テキスト画面のVRAMを知る	70
	ディスプレイエリア (表示用)と	
	アトリピュートエリア (装飾用)	70
3	HE 2000	
	キャラクタ・コード	-
_	かなやアルファベットも数値で扱う	
4	> 1 > C = 1 - > > > > > = 0	
	属性を変化させる 使い方あれこれ	
5		
	エリアを操作する	
	文字"A"の属性を変化させる	
	文字をリバース (反転) させる	
	リバースに使用したマシン語命令	80

	プログラムリストの見方について84
	1行ブリンク (点滅) させるには84
	1行を消してしまう~シークレット86
	1行にいろいろな属性をもたせる87
6	カラーモードでアトリビュート
	エリアを操作する92
	サブルーチンを使ったテキストのカラー化95
	バックグランド・カラーを変える97
	ブロック転送命令を使う97
7	TEXT文字とバックグランドの
-	カラー化デモプログラム100
	デモプログラムの仕様100
	全体のキャラクタに色をつけるには102
	バックグランドに色をつけるには103
	プログラムはこうなっている103

第3章	キーボードを操作する -キーボードからデータを読み取る
1	キーボードetc.に使われる //の命令113 //oボトから8ビットデータを
2	取り込むIN命令·······113 I/Oボートにデータを出力するOUT命令·······114 キーボードから

ゴーク も 詩 フ・レ ス に ノナ
データを読みとるには116
ビットを調べればキー入力がわかる116
対応ビットが 0 なら、キーは押された118
キー入力のテストプログラムを作る119
論理演算命令を使って
プログラムを作る121
論理演算命令は、大きく3種類121
かけ算みたいなAND命令······121
たし算みたいなOR命令124
等しければ 0, XOR命令 128
1ビットごとの論理演算で
ドット単位の移動が可能に132
論理演算の特殊な計算133
キー入力のテストプログラムを作る135
キー入力とキャラクタの
移動プログラムを作る137
キー入力とキャラクタの
4 方向移動プログラムを作る137
8 方向の移動プログラムを作る145
絵描きプログラムを作る150

グラフィックスの世界

~グラフィックス・バターンの作り方と移動方法

1 グラフィックスの原理………159

2	ビット・マップ法によるグラフィックスグラフィックス・パターンをカラーにすカラープレーンを切り替えるには・グラフィックス・パターン"花"の表示・・・CRT中央に "花" を表示するプログラムの作成・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	るには…161 163 165 165 166 170
	ヘリコプターの目を凹転させる	
	スムーズ・スクロール	
ij.	スムーズ・スクロール ー横方向をドッ	ト単位で動かす
1		
1	ー横方向をドッスムーズな動きの原理 右へシフトするには問題がある	195
1 2	ー横方向をドッスムーズな動きの原理・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	195 195
1 2	ー横方向をドッスムーズな動きの原理 右へシフトするには問題がある	195 195
1 2	ー横方向をドッスムーズな動きの原理・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	195 195 200
1 2	ー横方向をドッスムーズな動きの原理 - 右へシフトするには問題がある - ヘリコプターを横にスムーズに動かす - ヘリコプターを表示させるには - タドット目をどう移動させるか - 途中で方向を変えて移動するとき	
1 2	ー横方向をドッ スムーズな動きの原理 - ボーシフトするには問題がある・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	
1 2	ー横方向をドッスムーズな動きの原理 右ヘシフトするには問題がある ペリコプターを横に スムーズに動かす ペリコプターを表示させるには 9ドット目をどう移動させるか 途中で方向を変えて移動するとき スムーズな移動のプログラム テキスト画面とグラフィックス画面を	
1 2	ー横方向をドッ スムーズな動きの原理 - ボーシフトするには問題がある・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	

マシン語アラカルト

1	()はどういうときに使うのだろう240)
2	マルチタスク可能な割り込み機能24	
3	先入れ後出しスタック・ポインタ242	,
	1 ●なぜSPが必要になるのか·····242	
	2 @SPはどのように動くのだろう243	3
	3 ●他の使用例CALLとRET245	
4	いろいろ使えるスタック操作命令246	i
5	SPはC000H以前に248	
6		,
7		
8	タンスのようなバンク切り替え254	
9	バンク切り替えに似たウィンドウ機能255	
10	メモリモードを切り替えるには258	
11		
12	自由にファンクションキーを	
	設定したい267	
13	VRAMを移動させる269	
14	ユーザが使える未使用命令 271	
15	プロテクトはずし274	
	1 ●モニタにさえ入れない274	
	2 ●Pオプション解除プログラムを作る275	
	3 ●Pオプションを解除せずにモニタに入る276	
85	一気に移動させるブロック転送命令279	
17	微妙な中間色も出せるアナログモード284	
18	キーワードと中間言語を見る292	
		_

付録

80桁×25行モードVRAM	番地表304
40桁×25行モードVRAM	番地表306
ROM内システムサブルー	- チン307
RAM上のシステムワーク	フェリア312
システム 1/0 ボート	315
Z-80 (Z-80A) = - ₹= ッ	ク
↔機械語対照表	326
10進数↔16進数変換表…	330
1 バイト符号付16進数・	331





第章

初級者から中級者までの

マシン語

誰もが知っておきたい基礎知識

「マシン語って難しい」 「いろいろ本を終んだけどよくわから なかった。 こんな活をよく聞きます。マシン語は他 の言語 (BASIC や FORTRAN) などとは較べ ものにならないほどレベル差の激しい言語です。ですから、レベルに合った知識 が必要で、3かなアの人は生でするいと彼の

ものにならないほとレイルをの感じい言語です。こですから、レベルに合った知識 が必要で、初めての人はまずマシン語の 大棒をつかむことが大切です。 そこでこの章では、マシン語を勉強し ていく上で必要な後小限の知識を1ンパクトにまとめてみました。初めての人は とにかくこの章をとしてくり洗んてくださ い、そうでない人し、ざっと自分の知識

を再確認して、そりないところを補って

2進数と16進数

2進数は0と1の世界

コンピュータの世界は、0と1の2通りから成り立っています。したがって、 コンピュータに命令を与えたりするときは、0と1だけの組み合わせ、すなわ ち2進数で行うことになります。■1-1に2進数と10進数の対応表を示します。

2 海数では0と1しかありませんから、1

桁で表せる数は()と1だけです。これが2桁 になると00,01,10,11の組み合わせがあります から、0、1、2、3と4つの数を表現でき るようになります。同じように3桁では8つ の数が表現できます。桁数が増えれば増える ほど、表現できる数は多くなります。一般に 桁数と表現できる数との関係は次の式で表さ 扎束寸.

2"=奏すことのできる■(n は桁数を表す) たとえば2桁のときは、

 $2^{n}: 2^{2}=2 \times 2=4$

で4個の数を表現で作ます。4桁のときは、 2": 21=2×2×2×2=16

で16個の数を表現できることになります。

第1-1 2律数と10年数

進数	10進数
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	. 7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
	1.5

■ビットとは2進数の桁のこと

2 進数の桁のことをピットと言います。PC-8801mkHSR は8 ピットマシン と呼ばれていますから、一度に扱える2進数の桁は8桁です。8ビットで表現 できる数は、

21=256

たので PC-8801mkHSR では、…麼に256個の数を表現できるわけです。 II ビッ

第 | 章 初級者から中級者までのマシン語

トマシンでは、この256までの数値を使用してコンピュータに命令しています。 なお、これからは2週数の桁のことをピットと呼んでいくことにしましょう。 ピットの数と表現できる数との順係を**奏1-2**に示します

表1-2 ビットの数と	1E91	→ 2	9ピット	→ 512
表現できる無	2ピット	. 4	○10Ey k-	• 1024
E○印はよく使 用されるヒッ	3ピット	. 8	11ピット …	→ 2048
1	4ビット	• 16	12ピット	→ 4095
	5ピット	→ 32	13ビット	→ 8192
	6ビット	- 64	14ビット	→16384
	7ピット	-+128	15ピット	32768
	8ピット	-256	016ビット	→65536

10選款では、一、ド、ドと格ごとに位かついていました。同じように、 2進数も桁ごとに位を持っています。この2進数の桁の位のことを重みと言う 場合かあります。8 ピットを倒にとって、桁の位を見てみると関いつようにな ります。これを見ると、2 進数を10進置に直すには、1 になっているピットの 位の重みを建していけば良いことかわかります。ピット数が多くなっても同じ ことです。

図1-1 13 -- 10 | 故に直す

2 進数 8 ビット	T	9		0	0 1	0	- 1	1	
2 進数の桁の位 *()の中は10進数の重み	(128)	(64)	25 (32)	2 ⁴ (16)	(8)	(4)	(5)	(1)	
10進数に直すと	1 × 27	0 × 2 ⁵	1 × 25	0×2*	0×21	8 × 22	1×21	1×2°	
	128+	0 +	32 +	0 +	0 +	0 +	2 +	1	= 163

ビットとバイト

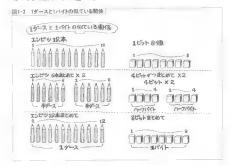
2 進数は0と1を使用しているので、よく10進数の数値と間違われます。そ こで10進数と区別するために、2 進数の数値の最後に"B" (Binary) をつけま す たとえば、

1001B (議み方 イチ, ゼロ, ゼロ, イチ) これは2進数ですから、当然、読み方も10進数とは違います、2進数には0と 1 しかないので、例のように「イチ、ゼロ、ゼロ、イチ」と読むことになります。しかし、ビット数が大きくなって8ビットや16ビットになると、何のことを言っているのかわからなくなります。これでは大変です。

■8ビット=1バイト

そこで、何ピットかずつ区切って、これをひとつの単位として扱うことが考え出されました。現在は8ビットをひとつの単位(1パイト)として扱っています。そして1パイトの半分、すなわち4ビットのことをハーフパイトと呼んでいます。昔は1パイトが5ビットとか6ビットとかまちまちでしたが、コンビュータが発達してからは、文字の記憶に便利な8ビットに定着しました。

8 ビットを 1 バイトと呼んでいるのは、エンビツ12本を 1 ダースと呼んでいるのとよく似ています(■1-2)。



16進数を使って2進数を読みやすく

8ビットをしバイトと呼ぶのはよいのですが、呼び方が「イチ・ゼロ、イチ ……」では今までと何ら変りありません。そこで、わかりやすくするために1 バイトを4ビット、つまりハーフバイトすつ反切って誘む方法があります。4

第 Ⅰ ■ 初級者から中級者までのマシン語

ビットで表せる数は全部で16です。そこで、16種類の文字からなる16進数を使って1パイトを流むとわかりやすいのです。

16周 は、0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、Fの16種類の文字から減り立っています。A、B、C、D、E、Fはアルファベットの文字ですが、16進数では数字の意味ですので注意してください。 厚住、W、Z、……などという文字も使用されていましたが、現在はすべてA~Fの文字に載しまれています。

■"H"をつけると16進数

また、16進数でも 0 から 9 までは10進数の文字を使用しているため、10速数 か16級数が区別できません。そこで16進数を10進数とはっきり区別して戻すた めに、2 進数のときと同じ方法で16進数の数値の最後に"H" (Hexadecimal) をつけています。たとえば、

8230H

となります。

さらに、16進数では0-Fまでの文字を使用しているために、A-Fが先頭にくる場合もあります。もし、A-Fが先頭にきたときに16進数の数値であることを強調したい場合は、A-Fの文字の前に"0"をつけます。

0F3C0H

この表記法は特にアセンブリ言語を便用したときに、ラベルと数値を区別するために使用されます。アセンブラにかけたプログラムリストを見るときや、

1-3 2進数,10量数, 16

2進数	10進 教	16進数
0000B	0	. DH
0001B	1	18
00108	2	2H
00118	3	3H
0100B	4	4H
0101B	5	5H
0110B	6	5H
0111B	7	7H
1000B	7 8 9	8H
1001B	9	9H
1010B	10	AH
1011B	11	ВН
1100B	12	CH
1101B	13	DH
1110B	14	EH
1111B	15	FH

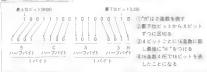
雑誌などのリストを見るときにでてきますね。

2 進数, 10進数、16進数をまとめると、P.18の表1-3のようになります。

■ 2 進数から16進数へ

16進数は4ビットデータをちょうど1桁で表すことができるので、 II ビット や16ビットをそれぞれ2桁、4桁で表現できます、2進数を16進数に直す場合、 図1-3のようになります。

図1-3 2進数を16運動に直す



2 進数で表すよりも16進数で表した方が、数値の大小関係がわかりやすいですね、この方が覚えやすいし、間違いも少なくなります。さて、こうなると2 差数など必要ないようですが、そうはいきませんが、2 進数ならぼすぐに おかります。PC-880ImklISRでは、グラフィックスにピット・マップ法(グラフィックスの壁で説明します)を採用しているために、2 進数が締めば必要と なってくるのです。2 進数と16進数の復習を簡単にやってみましたが、いかが でしたか? まあ、おれていたひとはもう一度読み直してください。

Z-80A CPU

Z-80Aは8ビットCPUの主流

8 ビット CPU を使用したパソコンは数多く発売されていますが、その多く で "Z-80A" という CPU が使われています。 Z-80Aは、8 ビット CPU の目途 といえるからしれませんね、CPU とは、中央演算処理装置(Central Processing Unit) のことで、コンピュータの頻繁にあたるもので、すべての計算は必ずこ の部分で行われます。

現在、マイコンの CPU は、

8080系 CPU (インテル社)

6800系 CPU (モトローラ社)

の2つに大きく分かれています。なお、8080という CPU は Z-80A(ザイログ社) に取って代られており、6800も6809 (モトローラ社) に取って代られています。 PC-8801mkIISR には、この Z-80A が使用されています。

Z-80A CPU は、40ビンのブラスチック(またはセラミック)の懐体の中に人っていて、その本体は $5\sim6$ mm四方の小さいものです。したがって、CPU ϵ 面接見ることはできません。

CPU の内部は見ることができませんが、プログラムに関することはマニュアルで知ることができます。プログラムに関係するレジスタは約20個しかありませんが、この約20個のために100種類を越す命令が用意されています。

Z-80Aの内部構成

マシン語を学ぶ際には、CPUの内部構成を覚える必要があります。CPUの内部構成は図1-4のようになっています。

それでは、CPU について簡単に説明していきましょう。まず、レシスタとは、 CPU の内部にある特別な働きをするメモリです。 CPU を構成しているレジス タは、それぞれ特徴を持っていて各々に名前がつけられています。各々のレジ スタとその働きをまとめてみましょう。

■主レジスタ・セット

自由に使用できるレジスタで、計算を行うなど用途がたくさんあります。

Hレジスタ	しレジスタ
Dレジスタ	Eレジスタ
Bレジスタ	Cレジスタ

A レジスタ F (フラグ) レジスタ

この中で最も大切なのは、Aレジスタ(アキュムレータ)とF(フラグ)レジスクです。Aレジスタは特別な機能を持っていて、すべての計算の中心になる部分なので特別にアキュムレータと呼ぶこともあります、次にF(フラグ)レジスタですが、これは汎用レジスタ(H. L. D. E. B. C. Aレジスタのこと)の状態を表すものです。プログラム上の分岐はすべてこのレジスタのお世話になるので、非常に大切なレジスタです。また、Aレジスタと「フラグ)レジスタをあわせて PSW(Program Status Word)ということもあります。

汎用レジスタは、それぞれ8ビットのレジスタですが、次のように組み合わ



■ | 章 初級者から中級者までのマシン語

せると16ピットレジスタとしても使用できます。

(上位パイト) (下位パイト)

Hレジスタ + Lレジスタ = HLベアレジスタ

Dレジスタ + Eレジスタ = DEペアレジスタ Bレジスタ + Cレジスタ = BCペアレジスタ

(8ビット) + (8ビット) = (16ビット)

なお、16ビットレジスタとして使用する際には、上記以外の組み合わせ (たと えばHレジスタとBレジスタ) では使用できません、16ビットレジスタとして の使用は、アドレス (番地) などに数多く使われます。

■補助レジスタ・セット(副レジスタ・セット)

この補助レジスタセットは、すべての面で主レジスタ・セットと全く同して す。補助レジスタ・セットは、主レジスタ・セットのデータの一時的な保存に 便用します。しかし、その順 CPU はキレジスタ・セットを使用しているのか、 補助レジスタ・セットを使用しているのか情報を出しません。そこで使用方法 を仕っきりきせておく必要があります。

▼ SP (スタック・ポインタ)

汎用レジスタなどのデータを逃避するときに使用されるものです。1回の命令 で2パイト分(つまり、16ビット・ペアレジスタとして)退避しますので便利 です。

▼ PC(プログラム・カウンタ)

次に実行する命令のメモリのアドレスが入っているところで、命令によって 自動的にセットされます。

▼ IX, IY (インデックス・レジスタ)

メモリを呼び出すための専用レジスタです。IX, IY は全く同じ働きをしています。

▼R (メモリ・リフレッシュ・レジスタ)

ダイナミック・メモリ (メモリのひとつで、多くのパソコンで使用) に使われているものです。 乱数の基数などに利用されることがあります。

▼ 1 (割り込みベクトル)

モード2という状態で使用されるベクトルで、このレジスタにはアドレスの 上位バイトが入っています。しかし、今回は使用しません。

▼ IFF(インタラプト・レジスタまたはインタラプト・フリップ・フロップ)

割り込み許可をするか、しないかを決めるもので、プログラム上で操作する ことができます。PC-8801mkHSR では CRT などに使用されていますので、こ のレジスタはプログラムによってコントロールする必要があります (割り込み についての説明は、P.241参照)。

CPU の内部構成を説明すると、以上のようになります。具体的な使用方法は プログラムを作りながら覚えていきますが、各レジスタの名前とピットの長さ は今ここで覚えておいてください。

マシン語命令の長さ

マシン語の命令の長さは一定ではなく、1 バイト命令から 4 バイト命令まで あります。特にハンド・アセンブルでアセンブリ言語をマシン語に変換すると きには、このマシン語の命令の長さが必要となってきます。ハンド・アセンブ ルを行うときは、必ず対応表で調べてください。マシン語命令のバイトの長さ をæすと、閏1-505 種類になります。



マシン語命令は、1パイト命令から4パイト命令まであります、Z-80A CPU は8ピット・マシンですから、1パイト命令を扱う際には問題はありません。 しかし、4パイト命令などは一度にCPU 内に取り込むことができません。そこ でCPUは、命令を8ピット(1パイト)ずつ4回に分けて取り込むことになり

■ | 章 初級者から中級者までのマシン語

ます。CPU は命令の長さに応じて取り込む回数を自動的に調整するので、取り 込む回数に注意する必要はありません。

■ハンド・アセンブルの際には上位と下位を運に

図1-5の 3・バト命令と 4・バイト命令を見てください。「下位データ」上位データと下位データの方が先に書いてありますね。これらの命令ではデータか2 パト・データなので当然 CPU (オテータの部分を 2 同に分けて取り込むことになります。このとき CPU の鰤が出て、2 バイト・データの場合は必ず下位データから先に取り込みます。そこで、ハンド・アセンブルの際には往途が必要です。

■ JP 8800H をハンド・アセンブルすると…

1-2714

8 8 0 0 H (2 バイト・データ)

上位データ TE 8800H (アセンブリ言語で書いたジャンプ命令)

これをハンド・アセンブルしてみましょう。

正しいハンド・アセンブル作業 誤ったハンド・アセンブル作業

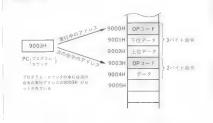
JP 8800H *上位パイト・データと 下位パイト・データを 逆にする JP 8800H C3 88 00

以上のようになりますので、注意しましょう、なお、アセンブラを使用した場合にはアセンブラ自身で上位パイト・データと下位パイト・データを逆にしてくれますので注意する必要はありません(ただし、ダンブリストを見るときは上位と下位が逆になっていますので注意してください)。

■ OP コードは命令の頭にく ■

OP コード (オペレーション・コード) は、必ずひとつの命令の中では一番前 になっています。この OP コードを解読することによって CPU は、命令が何バ イト命令であるかを理解するのです。

先程の CPU の内部構成の項で PC (プログラム・カウンタ) というものが出てきましたね。 PC (プログラム・カウンタ) は次に実行する アドレスを示すものでした。 次に実行する アドレスとは、次の命令の OP コードのある アドレス のことです。 たとえば現在の命令が 3 パイト命令であるならば、次の OP コードのある アドレスは現在実行中の アドレスの3 つ先ということになります (図) ー6)、したがって、/モリの割り当てを問題えると、CPU は単なる データを OP コードとして取り込みますので、 勝退ということがおこります。



Fレジスタを探る

コンピュータがすばらしいという別由のひとつに、ある条件に従って判断して て(れるというのがあります。BASIC 言語では、「IF~THEN」などの命令に よって判断を行っています。この制かのもとになっているのが、CPU の中にあ るF(フラグ)レジスタです。マシン語を理解するには必ずこのF(フラグ) レジスタの意味を理解しなければなりません。

■溶質結果の状態を表す旗

マラダ (Flag) とは繋のことです。よく運動会などで、赤駄と自様を上げた り下げたりしてスタートの準備ができたかどうか信分を送っているのを見かけ ると思います。駅の上げ下げは準備軟況を表しており、それを見て進行係は襲 技を進行をせます。これと同じことが、CPUのフラグにもいえます。

フラグは、CPUが演算を行った結果こういう状態になった、ということを教えてくれるものなのです。

CPUのフラグとは、F(フラグ)レジスタの各ピットのことです、Fレジス タだけは他のレジスタと違って、データなどをロードしたりしません、CPU は 領籍命令を実行すると、ある種のデータをレジスタに書き込んでくれます。私 たちはこのFレジスタのデータを見ることによって、演算結果の状態を知るこ

第1章 初級者から中級者までのマシン語

とができるわけですが、一般にマシン語のプログラムではFレジスタのデータ を見る必要はありません。

今戦命令 (たとえば、「JP Z」や「CALL C」など) ではフラグが立つ |フ ラグが!になること), 立たない (フラグからになること) ということを CPU | 小型 (大きない) (フラグからになること) ということを CPU | 小型 (大きない) (マラグルで) (マラグルで)

■Fレジスタの各ビットの意味

Fレジスタの各ピットの意味は図1~7のようになります。



▼ゼロ・フラグ

ある命令の実行結果、アキュムレータが0になったときにセット(1が立つ) されます。アキュムレータが0でないときはリセット(0になる)されます。 よく使用され、同じ値であるか調べるときなども使用されます。

▼キャリー・フラグ (言憂)

- キャリー・フラグは次のような結果が生じた場合にセットされます。
- 加票の結果、桁上りが生じた場合

減算の結果、桁下りが生じた場合

ローテート、シフトの場合

上記のような条件が立たなければ、キャリー・フラグはリセットされます。また、キャリー・フラグは演算の他にアキュムレータのビットを調べたりするの にも使用されます。

▼サイン・フラグ

符号付数字を扱う場合に使用され、演算結果が負 (マイナス) の場合にセットされます。

▼バリティ/オーバーフロー・フラグ

このフラグは2つの機能を持っています。そのひとつは、途埋演算を実行したときに、その結果アキュムレータ内で「1」を示すビットの個数が偶数ならばパリティ・フラグがセットされ(1が立つ)、奇数ならばリセット(0になる) される機能です。

もうひとつは、符号付2の補数演算の結果オーバーフロー(桁あふれ)が生 とたときにオーバーフロー・フラグがセットされる機能です。

▼ハーフキャリー・フラグ

アキュムレータの下位4ビット目から桁上りまたは桁下りがあったとき、セットされます。10進数の補正命令(DAA)を実行するときに、このフラグが用いられます。

▼波算フラグ

BCD (2 進化10進) 演算を補正するアルゴリズムは加算と減算とで異なります。そこで、DAA 演算が止しく補正処理を行えるようにこのフラグが用いられます。 最後に実行された命令が滅算ならセットされ(1になる), 加算ならリセット(0になる) されます。

以上、フラグ・レジスタの各ビットについてまとめてみました。この中で、 特に賞える必要があるのは、

ゼロ・フラグ

キャリー・フラグ

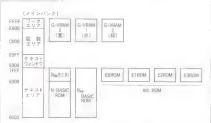
の2つです。その他のフラグは、必要になったときに覚えてください。

PC-8801mkIISRの ハードウェアを知ろう

メモリはどう使われているか

BASIC と違ってマシン語では、常にメモリの使い方を管理する必要があります。メモリマップ (メモリの使われ方) を理解していないと、マシン語のプログラムは作れません。本書では、メイン・メモリ以外のメモリも操作することになりますから、メモリマップをよく覚えておいてください。では、PC-8801 mkHSR のノモリマップを図ー6に示します。

図1-8 SRのメモリマップ



■バンク切り替えで212K バイトのメモリ空間

■ピット・マシンでは、一度に0000H 番地から FFFFH 番地までのメモリし か特定できません。すなわち、一度に持てるメモリ空間は64K バイトまでとい うことです。しかし、PC-8801mkIISR ではパンク切り替えという特殊な方法を 使用しているので、212K バイトのメモリ空間を持つことができます。 65ろん、メイン・メモリ以外のメモリを指定するときはアドレスを 2 同に分けて指定しなければなりません。 具体的な操作については、検査で練習しながら学んでいきましょう、特に、VRAM $0 \sim 2$ までのグラフィックス用のメモリは、グラフィックス両値やグラフィックスのカラー化に欠かせないものです。

なお、マシン語では主に、メイン・メモリと VRAM $0\sim 2$ までを使用しています。

システムからVRAMまでコントロールするI/Oアドレスマップ

前節ではメモリマップについて説明しましたが、ここでは、メモリのパンク 切り替えを初めとして、すべての動作をコントロールする I/O アドレスマップ について説明していきます。

CPU (Z-80A) は、メモリをコントロールする以外に特別に外部へ入出力するアドレスを00HーFFH までの256帯拠待っています。PC-8801mkIISR では、この外部へ入出力 (INPUT/OUTPUT) するアドレスをうまく使用して、メモリを増設させたリシステムをコントロールしたりするのです。したがって、マシン語では1/O アドレスマップを終対に知る必要があります。1/O アドレスマップを表出に派にます。

264.1	- 4	1/0	175	145	7.77	T

1/0アドレス	機 能 内 客					
00H~0BH	キーボード・マトリックスで、キーボードからの読 み取りのときに使用される (第3章参照)					
10H	ブリンタのDATAボート、プリンタへのデータがセットされる番地					
20H, 21H	カセットやRS-232Cインターフェースをコントロー ルする					
30H, 31H, 32H	システムポートで, システムをコントロールできる 重要な番地					
34H, 35H	G-VRAM制御					
40H	ブリンタやRS-232Cのタイミングをコントロールする					
44H, 45H	プログラマブル・サウンド・ジョネレータ(FM音源)の 制御					
50H ~5FH	CRTコントロール、画面に関してすべてコントロールできる重要な番地					

第 | 初級者から中級者までのマシン語

60H 68H	DMAコントロール(CPUを通さずに直接、メモリと 周辺機器の間でデータを転送することをDMAと言い、 CRTとVRAMの圖で使用されている)
70H, 71H, 78H	メモリコントロール. 拡張ROMなどのコントロール
80H -8FH 90H -9FH A0H - AFH B0H - BFH C0H -CFH D0H ~ DFH	拡張用スロットパス. 自由に使用できる
E2H, E4H, E6H	主に割り込みコントロール、インタバルタイマーな どのコントロール
E8H ~ EBH	漢字ROMコントロール
FCH ~ FFH	フロッピィディスクコントロール

どうです? わかりましたか? これを見ただけでは、何をどのようにコントロールしているかわかりませんね、後季でプログラムを作りながら説明していきますので、そのときまでお待ちください。というのは、ここで説明してもプログラムがないと理解しにくいからです。ですから、ここでは1/0 アドレスマップが何をコントロールしているかど1世解してください

N₈₈-DISK BASIC使用時のメモリマップ

マシン語のプログラムは、RAM 領域であればどこからでも作ることが可能です。しかし、実際にはどこからでもというわけにはいきません。その理由は Near-DISK BASIC を立ち上げたときには DOS(Disk Operating System; アイスクのファイル管理プログラムが動作しているからです。この DOS をうまく 使用しないと、ディスクにプログラムを"セーブ"、"ロード"できません。DOS をマシン語のプログラムで載むことも可能ですが、まず著人では無理でしょう。したがって、マシン語のプログラムエリアは事実上。DOS の部分のプログラムエリアを使用することはできません。

■マシン語プログラムエリアは B900H 番地から

そこで、DOSのエリアを壊さない範囲でマシン語のプログラムエリアを決定 しなければなりません、DOSのエリアはオープンするファイル画や変数によっ てその都度変ってしまうので、安全性をとってファイル数を1にしたときのマシ ン語プログラムエリアを B900H 系地からとします(**図1-9**)、DOS のエリアを 壊していないので、 N_{st} -BASIC コマンドの "BLOAD. BSAVE" が使用でき、 モニタコマンドを使うよりずっと楽にマシン語プログラムのファイル管理ができます。

田途半端で、B9001 番地からマシン隔ブログラムエリアを始めている理目は、他にもあります。図1-3を見ればわかるように、グラフィックスのためのブレーン (G-VRAM) が C0001 番地から妨まっているために、プレーンを操作する関係しどうしても C0001 番地に前にマシン満プログラムエリアを設定したいからです。第4歳(P.163)で説明してありますが、グラフィックスのプレーンを操作するためにはバンク切り替えという操作を受要するので、C0001

図1-9 Nati DISK BASIC使用時のメモリマップ

1000Н	N ₈₈ - BASIC インタブリタ ROM	テキスト エリア マシン語 ブ ログラムエ リアとして 使用可能			
3000H	テキストウェンドウ		_		
3900H	DISK Code ファイルバッ ファ等	DOSのフログラムの使用エリア (このエリアは使用してはならない) CLEAR、&HBBFFで指定する (Na=BASICのコマンドを使用するとき)			
C000H	マシン語 プログラム ' エリア (本書で使用)		G- VRAMO	G- VRAM1	プレーン2 G- VRAM2
Е600Н	Nss-BASIC				
F3C8H	ワークエリア (VRAM)	SR# VRAM			

第 | 章 初級者から中級者までのマシン語

番地以降にバンク切り替えがなければなりません。

B900H 番地から E5FFH 番地までのメモリの大きさは10K バイト近くあり ます、これは、データを大阪に扱うプログラムを作るのでなければ充分な大き さです。もし、これ以上の大きさが必要となったら、別のテキストエリアを使 用すれば扱いのですが、未書ではそこまで長いプログラムは作りません。

以上の理由から、特別の指示がない限り、マシン語ブログラムエリアを

PC-8801mkIISR B900H~E5FFH書地

とします。

■モニタに入る前に

ただし、BASIC コマンドの「CLEAR 命令」でマシン語エリアの宣言をして いないので、BASIC モードに戻ったときに BASIC 命令が100%使用できませ ん、もし、完全に使用したいのであれば、必ずモニタに入る前に次の命令を実 行してください。

CLEAR, &HBBFF

これで、すべての命令が使用できます。しかし、拡張命令のエリアとマシン語 プログラムエリアが同じなので拡張命令(CMDのついた命令)は使用すること ができません。

マシン語モニタを使いこなすために

非常に強力なマシン語モニタ

マシン湖の入力は BASIC でもできますが、本格的に入力しようと思ったら、マシン湖モニタの対世部にならなくではなりません。PC-8801mkHISR 用のモニタは、パソコン用としては非常に強力なものと思われます。特長は次のようになっています。

▼特長

- ①※ミニアセンブラ機能を持っています。
- ② 数値を16進数、 B 進数のどちらでも入力、表示が可能です。
- ③ 1/0 ボートへの入出力が可能です。
 - ④※逆アセンブル機能を持っています。
- ⑤ CPUのレジスタの圖の表示, 変更が可能です.
- ⑥ カセット、フロッピィディスクにロード。セーブが可能です。
- ② スクリーンエディタによるメモリ内容の変更ができます。⑧ Help コマンドにより、コマンドの入力形式がわかります。
- (NEC 3-#- x 7 = 3 7 /L 11)

■だけどまだまだ不満がある

これだけの背長を備えていながら、①と①については非常に不満です。 なぜなら、このミニアセンブラは Z-80A という CPU のニーモニック (ザイログ社) を採用しているからです。したがって、このミニアセンブラを使用する場合は、Z-80A のでもンブリ 言語は使用できず。8080A のアセンブリ 言語は使用できず。8080A のアセンブリ 言語は使用できず。8080A のアセンブリ 言語は使用できず。8080A のアセンブリ 言語は使用できず。2080A のでは、ニーモニックを 2 種類覚えなければならず、ユーザに負担がかかってしまいます。そのに、Z-80A が持っている強力な命令などは受け付てくれないので、ハンド・アセンブルする必要があります。■1-5に Z-80A と8080A のニーモニックの対応変を掲げておきますので、このミニアセ

第1章 初級者から中級者までのマシン語

ンプラを使用したいときに利用してください。

次に④の遊アセンブル機能ですが、これも不識です。というのは、遊アセン ブラとミニアセンブラとは好になっているからです。 つまり、ミニアセンブラ では8080A アセンブリ : 請しか使用できないということは、遊アセンブラでも 8080A のアセンブリ : 言語しか使用できないこと 意味するのです。

■1-5 8080AとZ-80CPU(Z-80A)の命令対応表(8080Aにないものは省略)

8080A	Z-80	8080A	Z-80	AG808	Z-80
ニーモニック	ニーモニック	ニーモニック	ニーモニック	ニーモニック	ニーモニック
ACI	ADC A. N	IN	IN A _s (N)	POP H	POP HL
ADC M	ADC A.(HL)	INR M	INC (HL)	POP PSW	POP AF
ADC r	ADC A. r	INR r	INC r	PUSH B	PUSH BC
ADD M	ADD A,(HL)	INX B	INC BC	PUSH D	PUSH DE
ADD r	ADD A, r	INX D	INC DE	PUSH H	PUSH HL
AD1	ADD A. N	INX H	INC HL	PUSH PSW	PUSH AF
ANA M	AND (HL)	INX SP	INC SP	RAL	RLA
ANA r	AND r	JC	JP C, NN	RAR	RRA
ANI	AND N	ML	JP M. NN	RC	RET C
CALL	CALL NN	JMP	JP NN	RET	RET
CC	CALL C. NN	JNC	JP NC. NN	RLC	RLCA
CM	CALL M. NN	JNZ	JP NZ. NN	RM	RET M
CMA	CPL	JP	JP P. NN	RNC	RET NC
CMC	CCF	JPE	JP PE. NN	RNZ	RET NZ
CMP M	CP (HL)	JPO	JP PO. NN	RP	RET P
CMP 1	CP r	.17	JP Z. NN	RPE	RET PE
CNC	CALL NO. NN		LD A.(NN)	RPO	RET PO
CNZ	CALL NZ. NN		LD A.(BC)	RRC	RRCA
CP	CALL P. NN	LDAX D	LD A.(DE)	RST	RST P
CP E	CALL PE. NN		ILD HL (NN)	RZ RZ	RET Z
CPI	CP N	LXI B	· LD BC. NN	SBB M	SBC A.(HL)
CPO		LXI D	LD DE, NN	SBB r	:SBC A. r
CZ	CALL Z. NN	LXI H	LD HL. NN	SBI	SBC A, N
DAA	DAA.	LXI SP	LD SP. NN	SHLD	LD (NN), HL
DAD B	ADD HL BC	MVI M	LD (HL), N	SPHL	LD SP. HL
			LD r. N	STA	LD (NN), A
			LD (HL), r		LD (BC), A
DAD H	ADD HL, HL	MOV M, r		STAX B	LD (DE), A
DAD SP	ADD HL, SP	MOV r. M	LD r, (HL)	STC	SCF (UE), A
DCR M	DEC (HL)	MOV rI, r2	LD r, r		
DCR r	DEC r	NOP	NOP	SUB M	SUB (HL)
DEX B	DEC BC	ORA M	OR (HL)	SUB r	SUB r
DCX D	DEC DE	ORA r	OR r	SUI	SUB N
DCX H	DEC HL	ORI	OR N	XCHG	EX DE, HL
DCX SP	DEC SP	OUT	OUT (N), A	XRA M	XOR (HL)
DI	DI	PCHL	JP (HL)	XRA r	XOR r
EI	E	POP B	POP BC	XRI	XOR N
HLT	HALT	POP D	POP DE	XTHL	EX (SP), HL

注, Nss-BASICモニタを使用する場合は、Z-80ニーモニックを8080Aニーモニックに変えて使用します。

Z-80A	モニタのニーモニック
JR (アドレス) JR Z, (アドレス) JR Z, (アドレス) JR NZ,(アドレス) JR C, (アドレス) JR C, (アドレス) DJNZ (アドレス)	JMPR(アドレス) JRZ (アドレス) JRNZ (アドレス) JRNZ (アドレス) JRC (アドレス) JRNC (アドレス) DJNZ (アドレス)

注1 Z-80Aの相対ジャンプカアドレスは符号付2の

類数を指します. 注2 モニタのアドレスは絶対書地 (ADOCH書地な ど)で指定します。モニタの中でこれを符号 付2の複数に計算してくれます。

※3 相対ジャンプについては確奪で説明します。

逆アセンブラとは、マシン語 (2進数もしくは16進数で書かれたもの)で書 かれたものをアセンブリ言語に直してくれるものです。使い方は、デバッグの ときにエラーが出た箇所のマシン語を逆アセンブルして、初めの命令通りアセ ンプルされているかどうかを剔べます。その他に、他人の書いたマシン語を解 析するとき16准数で書かれているものは解析しにくいので、逆アセンブルにか けてアセンブリ言語に直してから解析すれば非常に理解しやすくなります。

■市画のアセンブラ(DUAD-88D)を使う

8080A のニーモニックでは、Z-80A の能力を100%生かしきれません。 したが って、も1.100% CPU の能力を生かしたいのであれば、このミニアセンブラを 使用せずすべてハンド・アセンブルするか、Z-80A 用のアセンブラを使用する 1.かありません。本書のリストはすべて、DUAD-88D (アスキー) という Zー 80A のアセンブラを用いて作成しています。もし、本格的にマシン語を覚えた いのであれば、高価(49.800円)ですが指はないでしょう。

なお、PC-8801mkHSR で DUAD-88D を使用する際は、BASIC モードスイ ッチを N88V1 にセットしてください。

NEC とマイクロソフトが、どんな理由で 8080A のニーモニックをミニアセ ンプル機能に用いたのかほわかりません。次期新製品はユーザのためにも、 CPIIにあったニーモニックを採用していただけたらと思います。

いよいよマシン語の世界

ます、BASICモードとマシン語モニタのプロンプトの違いについて説明し ておきましょう。 プロンプトの違いは、このようになります。

OK-------BASIC が扱える!!!

h] または a] ……マシン語が扱える

第 | 童 初級者から中級者までのマシン語

プロンプトは人力モードであることをユーザに表示するものです。したがって、 どのプロンプトが表示されているかによって、どのモードであるかユーザが判 断することになります。

■モニタを起動する

次にマシン格モニタの起動と BASIC モードへ戻る方法について説明しましょう。

① BASIC モードからマシン語モニタを起動する



②マシン語モニタから BASIC モードへ戻る方法



マシン語モニタ

マシン語のコマンドは全部で22種類あり、使い方によって非常に便利なもの もあります。モニタコマンドの種類をまとめると、次のページの■1-7のように なります。

■モニタ使用にあたって

マシン語モニタを使用するにあたっては、多少の注意が必要です。ユーザーズ・マニュアルにはカセットテープについての規則も挙げてありますが、ディスク装置がついているのにかざわざカセットテープを使用することもないので、ここでは挙げません。ここでは使用上の規則として、数値入力、計算機能、側面形式の3つを挙げて注きます。

§4 マシン語モニタを使いこなすために

表1-7 マシン語モニタコマンド

コマンド	※ 味	機動
A	アセンブル	入力した1行をアセンブルします.
В	ベース	数値の表現形式を変えます。 (8進⇔16道)
D	ダンブ・メモリ	メモリの内容をディスプレイに表 します。
E	エディット・メモリ	スクリーン・エディタの機能を用 てメモリの内容を変更します。
F	フィル・メモリ	メモリの内容を定数で埋めていきす。
G	- II-	ユーザプログラムを実行します。
	インブット	1/0ポートの値を読み込みます。
L.	ディス・アセンブル	機械語を逆アセンブルします。
М	ムーブ・メモリ	ある範囲のメモリの内容を他のア レスのメモリ領域へ移します。
0	アウトブット	1/0ポートヘデータを出力します。
Р	ブリンタ・スイッチ	プリンタへの出力をコントロール ます。
R	リード・テープ	カセットテープからデータをロー します。
S	セット・メモリ	メモリにデータをセットします。
TM	テスト・メモリ	メモリをテストします.
٧	ベリファイ・テーブ	カセットテープの内容と、メモリ 内容を比較し直す。
W	ライト・テーブ	メモリの内容をカセットテーブに ープします。
х	イグザミン・レジスタ	CPUのレジスタの値を調べ、変更ます。
HELP # fold CTRL-A	ヘルプ	コマンドとそのバラメータの形式 ディスプレイに表示します。
CTRL-B	リターン	BASICモードへ復帰します。
CTRL-D *	ダンプ・ディスク	フロッピィディスクの内容をディー プレイに表示します。
CTRL-R *	リード・ディスク	フロッピィディスクから、データ· ロードします。
CTRL-W *	ライト・ディスク	フロッピィディスクヘデータをセーブします。

がついてる3つのコマンドはN-BASICでは使えません。

■数値は16進載で入力

PC-8801mkIISR は、16進数と8進数の両方を扱うことができますが、現在では16進数が定着し、8進数を扱うことはあまり多くありません。と言うよりは、 ほとんど使用しません。もし扱うとすれば、CPUのOPコードの解析に多少限 利なくらいです。それな程度ですから、8進数のモードは必要ないでしょう。

・16進数モード

アドレス 4桁:0000H~FFFFH

データ 2 桁: 00H~FFH

参考

8 進数モード

アドレス 6 桁:0000000~7777770

データ 3 桁:0000~3770

アドレスもデータも最後から4桁、2桁が有効になりますので、途中で入力 ミスをした場合はその時点から正しいアドレス4桁なり、データ2桁なりを再 よれできます。もちろん、16課数の配替である日はつける必要がありません。

注… D は 8 准数を表す記号 (Octal)

■便利な 機能

コンピュータでは16進数を利用しているために、暗算でたし算や引き置などができません。たとえばアドレスの計算などは一度10進数に実施し計算し、また16進数に直すということになります。BASIC モードにしなければならないので不便です。しかし、モニタの各コマンドレベルで加減(ただし、かっこや業実は使用できません)ができますので、アドレスなどの計算に減力を発揮してくれます。

例、F3F0H 番地に50H をたした結果のアドレスを求めた場合

h] SF3F0+50

F440 00

→求めたアドレス

■モニタに入る前に "WIDTH 80, 25"

マシン暦モニタに人るときは、BASICの"WIDTH"命令がそのまま引き がれますので、画面モードを決めるには、モニタに人る前に

WIDTH 80, 25

と命令しておいた方が便利です。マシン語でも画面モードを設定できますが、 BASIC で行った方が楽でしょう。 のちに説明するEコマンドは、テキスト画面のスクーロールウィンドウの影響を受けることがありますので、モニタに入る前に次の命令を実行して下さい。 CONSOLE 0、(17以上の値)

各コマンドはこう使う

■ A コマンド (Assemble=アセンブル)

- 格納開始番地 (ここでは D000H 番地)

このAコマンドはミニアセンブラで、インテル8080ニーモニックで入力した 1行をアセンブルし、できたマシン語をメモリに格納します。もし、Aコマン ドを使用するときに格納番塊を指定しない場合は、前に実行したAコマンドの かの寒地とかります。実行してみましょう。

A 開始—

h]AD288 D898 助一^{*}ジルの位置

▼画面 1

▼両面 2

Aコマンドで 俗納開始番地を D000日と指定して [・] キーを押す と、D000日 番地が表示され、カー ソルは15個のスペースを空けた位 窓で、ニーモニックの入力待ちに カリます

h1AD000 MVI A.30

ニーモニックを入力 (Z-80用では、LD A、30H) します。30は 16差 mなのですが、プロンプト(h])か46進数人力となってますので、16進数を示す"H"はつけていません。

h)AD800 D000 3E 30 MVI A.30 D002

***188** 3

ニーモニックを入力して ≥ キーを押すと、スペースを空けておいた同じ行にアセンブルされたマシン語が表示され、即、指定され

第1 初級者から中級者までのマシン語

た番地からメモリに格納します。 このニーモニックは2バイト命令 であることがわかります。次の人 力待ちになります。

このマシン語のプログラムは意味のないものですが、このような 歯面になります。 (NS) キーが使 用できますから、間違えて入力した場合などは入力した文字を訂正 できます

▼モニタに戻るには

Aコマンドのアセンブラからモニタへ戻るには、次の3つの方法が使用できます。

- ① STOP キーを入力する
- ② CTRL キーと C キーを同時に入力する
- ③ アドレスが表示されてニーモニックの入力待ちのときに 2 キーを入力する(画面 | のときや、画面 4 の D007H 番地が表示されているとき)

99 h] h] BQ 🕗

--- -- 入力する数値を 8 准数にする

数値をキーボードから入力したり、CRTに表示する場合に、16進数機いにするか、8.進数(Octal)扱いにするかを設定します、Nas-BASIC からモニタへ切り替えたときは、16進数モードになっています。8.進数は現在のところ、あまり催い道がないようです。プロンプトの遠いは、

8 進数 → q]

となります。では、さっそく実行してみましょう。

hJBQ qJm	
	ŧ
	K

* 1

16進数モードになっているの を、"BQ"を入力して8進数モー ドに切り替えた状態です。

-1917		
g]8H h]■		

▼ ■ 2

8 進数モードになっているもの を "BH" を入力して16進数モード に切り替えた状態です。

■Dコマンド (Dump=ダンプ)

入 力 形 式 D 表示開始番地。表示終了

例 L h] D0000, 00FF 2 (0000H 番地から00FFH までのメモリの内容を表示

します)

例 2 b] D0000 2 (0000H 番地から16バイト分表示します)

例 3 h] D. D3FF (前に実行されたDコマンドの最後の番地から, 03

FFH 番地までの内容を表示します)

例 4 h] D 🕗 (前に実行された D コマンドの最後の番地から16バ

イト分の内容を表示します)

Dコマンドはメモリの内容を CRT に表示するコマンドで、マシン語のプログラムやデータなどがよっていますので、これらを参照するときに使用します。 このコマンドで便利なのは対応するアスキー文字に記確には PC-8801mkIISR のキャラクタ)を右側に並示してくれますので、マシン語などのプログラムの中にあるメッセーン等を使す場合に利用できることです(ただし、16進数モードのみ)、では、実行してみましょう。

hines	100.	BBB	F														
6686	F3	31	AØ	E1	C3	E5	3B	96	7E	E3	BE	23	E3	C.5	93	6.3	
8818	23	7E	FE	3A	DB	C3	15	8A	F5	CD	42	ED	C3	25	59	6.8	
9828	7C	92	CØ	7D	93	09	88	88	3A	44	EC	87	C5	83	28	C9	
6836	3A	BD	EA	FE	88	C3	27	15	C3	69	E6		69	100	88	8.0	
8848	AB	21	FD	21	7E	A7	C8	FE	28	C9	21	FF	FF	22	56	E6	
8858	23	C9	22	41	EC	Cl	Cl	4F	78	D3	71	79	CI	C3	E9	11	
	CD	BD	85	CD	21	4F	C3	100	4E	CD	207	ED	F3	3A	C2	E6	
8978	C9		3A	B9	E6	A7	37	C4	21	500	C3	6A	6F	CD	96	18	
8888	F5	7A	FE	82	28	81	15	FI	CS	99	79	79	7C	7C	7F	58	
9898	48	30	32	28	7A	78	3E	22	98	88	AØ	21	56	22	14	22	
BBAB	24	24	10	24	53	25	29	26	99	21	E9	1 D	E6	10	53	1F	
SEBS	B7	IF	34	21	38	23	2F	23	5A	23	41	13	5F	21	06	90	
SUCE	SE	70	DE	m	67	78	DE	98	47	3E	88	C9	88	99	88	35	
RADE	JA.	CA	99	39	10	76	98	22	95	83	98	BA	DĐ	47	98	53	
BREB	Di	99	100	BA	18	9F	668	65	BC	CD	98	D6	77	3E	98	52	
2856	C7	4F	88	86	#B	100	1000	86	9B	101	03	86	88	86	88	86	
BBID	67	41	90	00	00												

▼西面1

Ngg-BASIC の ROM の表示

```
h108160.0205
8158 FF FF 81 88 88
                    91 81
                          88 81 FF
                                    FF 88 86
        19 86 28 86 88 FF
                          88 88 88
                                       88 88
                                             88 22
9188 88 PE
           C8 F3 66 66 66 86 86 68 CD
                                    EF 88
                                          88
6198 99 68 68 88 88 88 88 68 68 68 88 88 68 88 68
81A9 88 MM III 88 88 88 88 88 88 88 FF 88 15
                                             CB EE 66
RIBB SC
                 28
                    22 88 88 88 88
                                   66 88 68 88 88 88
                                                             load "
                       88 88
                             88 88
                                    88 88 88
                                             88 89 86
                                                            auto
0100 67
                       88 88
                             98
                                 818
                                    88 88
                                          88
                                             88 88 88
                                                            go to
81E8 6C
        69
                 28
                    8.8
                       86 86 98 88
                                    68 88 88
                                             88 88
                                                   98
81F8 72
        75 SE 8D 88 88 88 88 88 88
                                    98 88 86
                                             88 88
                                                   98
0208 73 61
                 28 22
                       68 88 88 88
                                    68 88
                                          9.0
                                             88 88
                                                             save
8218 58 65 79 28
                 88 88 88 88 88 88
                                    86 86
                                          68
                                             88 88 88
8228 78
           69 6E
                    28 88 88 66 88
                                    OR OR OR OR OR
                                                            print
8238 65
           69
                 28
                    2E 80
                          88 88 88
                                    86 88 88 88 88 88
8249 53
                    88
                             88 88
                                    86 88
                                          88
                                             20 20 20
8258 SC
        2752
                             9.9
                                 88
                                             66 68 68
8268 68
                                 2.0
                                             68 68
                                                   88
        58
          54
              28 65
                    6F 61
                                    28 28 28
                                             N9 88 88
#280 63 SF
           78 79 28
                    62 75
                                       68 88
                                             48 88 88
                                                            copy buffer
8298 4C
        50 54 20 66 65 65
                          64 88 88
                                      88 88
                                             88 88 88
8248 C3
        88 88 C3
                 88
                    88 FF
                                             3A C2
                                                                   AA FO 1861 579
8288 F5
        E6 FB D3 31
                    32 C2 E6 CD
                                                            #* E129*1-38#E12
                                                31 32
82C8 C2 E6 F1 E1 FB C9 E5
                             88
                                 36
                                    18 E8 E5 21
                                                            THE ALER ALCA
02D0 18 DA E5 21 79
                                                             Unive
```

▼画面2

N₈₈-BASIC でのファンクションキーの指定内容は、01B0H 番地から0244H 番地までで指定してあります。しかし、ROM の中に指定してあるので変更できません。

▼データをチェックしたいときなど

P コマンド (P.49参照) でブリンタ・スイッチがブリント・モードに指定されていれば、タンプされた結果がブリンタへも出力されます。ただし、人力形式が表示されないので、画面内に入るものだったら COPY キーを使用した力が良いでしょう。

また、Dコマンドを実行中に、両面表示を一時停止してデータをチェックも たいときなどは次のようにします

画面上の表示を一時停止させる………CTRL + S

■Eコマンド (Edit=エディット)

入力形式 E III

例 h] ED000 🗸

ーエディットの開始着地を DODOH 基地とする

Eコマンドは、スクリーンエディタの機能を利用して、CRT に表示されたメ モリの内容を変更するコマンドです。このEコマンドは開始番地を指定すると 自動的に256パイトの内容を表示し、さらに左側にはその1パイトに対応するア スキー文字を表示します。

▼メモリ内無を変更するには

メモリ内容を変更する際には、修正したいデータのところまでカーソル移動 キー(『旦ビー》やロールアップ、ロールダウンキー(ROLL)を使 用してカーソルを移動させ、新しいデータを入りします。新しいデータが入り されると、メモリの内容は変更され、左側に表示されているアスキー文字も変 配になります。このEコットは、マシン語のプログラムを入りしたり、一部 変更したりするのに便利です。

なお、使用できる編集キーは、

TOTAL ROLL BOWN

で、モニタへ戻るときは、

STOPまたは ESC を押します、

h1ED	000	1	0.0	0.0	00	00	9.0	88	99	9.0	ΘЯ	88	88	88	66	
D000	- 33	80	0.0	9.6	80	00										
D010	08	1000	98	9.6	0.0	90	90	00	00	80	00	80	88	0.0	00	
0828	88	1000	8.8	100	500	99	00	00	00	88	88	88	00	00	80	
0838	88	100	0.0	88	130	100	88	0.0	80	0.0	80	8.6	00	00	0.0	
D949	1000	88	00	00	9.8	0.0	00	00	00		98	0.0	9.0	99	9.6	
0858	9.0	03	08	88	08	88	919	0.0	553	00	88	00	98	00	90	
0868	88	9.9	88	100	80	90	88	88	80	98		99	88	0.0	0 B	
0078	8.8	103	88	9.0	100	88	88	88	00	9.6	00	88	80	88	00	
DRSR	-	88	103	100	88	88	88	0.6	8.0	100	00	88	200	80	00	
DRSR	-	88	00	8.8	88	88	100	EE	88	00	88	88	10	88	80	
DEAB	88	200	80	9.6	100	00	00	88	00	00	00	39	90	99	90	
DØBØ	00	100	8.0	98	100	00	100	8.8	00	00	00	90	88	0.0	98	
DOCO	919	88		88	00	153	00	900	0.0	00	88	00	00	88	88	
DODO	98	88	III	100	00	88	88	0.0	00	00	88	88	80	88	0.0	
DREB	00		88	8.6	88	8.8	100	98	00	0.0	88	00	99	00	0.0	
DREB	80	88	99	88	80	68	88	6.6	88	98	99	88	0.0	88	0.0	

▼■面

ED000 こをしたときの画面で、カーソルは D000H 番地に来ています。

■ F コマンド (fill=フィルメモリ)

■ h] FD000, D0FF, 00 🖸

開始番地 終了番地 セットする数値

第1 初級者から中級者までのマシン語

Fコマンドは指定した番地から番地まで、指定した数値をセットするコマンドです。このコマンドは、メモリの一部をクリアしたり、メモリの一部をある 定数にセットしておき禁走をくい止めたりするときなどに使用します。

```
blfD000.D0FF.FF
hlDD000.D0FF
NAME OF SELECT S
DASA FE FF FF
                                                 FF FF
                                                                            EF FF FF FF
                                                                                                                        EE EE
                                         EF EF EF
D070 FF FF FF
                                                                            FF FF FF FF
                                                                                                                FF FF FF
                                                                                                                                         EF FF
DORR FF FF
                                FF
                                         ER ER ER ER
                                                                            FF FF FF FF
                                                                                                                FF FF FF FF FF
D898 FF FF FF
                                                                            EF FF FF FF
                                                                                                                EE EE EE EE EE
DOAG FF FF FF FF FF FF
                                                                            FF FF FF FF
DOBO FF FF FF FF FF FF FF
                                                                           FF FF FF FF FF FF FF FF FF
DØCØ FF FF FF FF FF FF FF
                                                                            FF FF FF FF FF FF
                                                                                                                                FF FF FF
DOES FF
```

▼画面

D000H 番地から D0FFH 番地までFFHをセットし、セットされたかどうか Dコマンドを使用して確認しています。

■ G コマンド (Go=ゴー)

入力圏団 G 実行開始番地,プレイク・ポイント・アドレス#1, プレイク・ポイント・アドレス#2

例 [h] GD000, D007 [(プレイク・ポイントが1つのとき)

例2 h] GD000, D007, D010 🖟 (ブレイク・ポイントが2つのとき)

例3 h] GD000 🖟

Gコマンドはマシン語のプログラムを実行させるコマンドで、BASICで言 えば RUN に相当するものです。このGコマンドは入力形式に記述してあるよ うに、プレイク・ポイントを2つまで設定することができます。

▼ブレイク・ポイントとは

プレイク・ポイントとは、マシン語プログラム実行中に指定した番地でストップしたい場合に設定するもので、マシン語プログラムをデバッグするときにこの番地 (プレイク・ポイント) を通過するかしないかを確認するものです。 ですから、通過する子定の番地を通過しなかった場合は、それ以前の番地にバグがあるということになります。 なお、プレイク・ポイントが2つまで設定できるので、サブルーチンがある ときに一方はメイン・プログラムに設定し。もう一方はサブルーチンに設定し て油れがどう変ったかなどを調べることができます。

プレイク・ボイントを設定すると、そこを通過した場合、アレイク・ボイントの高地で実行を中止してモニタに戻ってしまいます。このとき CPU のレジスタの値は保存されていますので、Xコマンド(P-53参照)で CPU のレジスタを見ることができます。ここで PC (プログラム・カウンタ)レジスタを見れば、粉定したプレイク・ボイントであるかどうか知ることができます。

DANA NA DBBI 00 DARS 99 DAR3 DØ84 DRGS 30 0006 00 D007 80 0008 870 NOP Dead คค DØØB 88 Desc DØØD 88 DARE B B DØØF RST

W 100 ft 1

NOP 命令は何もしない命令 で、ただ通過するだけのものです。 RST7 にキニタへ戻るための命令 です(88に限る)。したがってこれ は、D000H 番地からプログラムを 実行すると、何もしないで D00EH 番地まで実行し、D00FH番地を実 行するモニタへ戻るプログラムで す。

hlGDBBB

A :88 F :PZ---E-- 8 :8888 8 :EDCC H :9881 A':88 F':PZ---E-- B':9808 D':FF7B H':8911 IX:2888 IY:7DCS | :F3 PC:088F SP:E5F9

ブレイク・ボイントを設定しないときは、PC レジスタはマシン語のプログラムを実行した最後のアドレスになっています。

h)GD889.D887

第 | 画 初級者から中級者までのマシン語

▼画面3

プレイク・ポイント・アドレスを D007H 番地とした場合, プログラムがブレイク・ポイントの D007H 番地でストップしたことが PC を見るとわかります。

higoses. DeeA

hJX A :88 F :PZ---E-- 8 :8800 D :EDCC H :8801 A::08 F':PZ---E-- 8':8800 D':FF78 H::8911 (X:2086 1Y:7DCS) :F3 [PC:D008] SP:ESF9

▼画面4

プレイク・ポイント・アドレスを D00AH 番地に設定しプログラムを実行させ たとき、やはり、プレイク・ボイントの D00AH 番地でストップしたことがわか ります

注意

プレイク・ポイントは、必ず RAM エリアにあり、プログラムが多分通過するであろう 帯地に設定しなければ効果がありません。

■ I コマンド (Input=インプット)

入力形式

| ポート・アドレス

例 h] 180 2 (I/O の8Dポートから 1 バイトのデータを読む)

1 コマンドは、直接ボート・アドレスから1パイトのデータを読み込んで、 CRT に表示するコマンドです。自作した1/Oポートをテストするときなどに 使用します。プログラムを根まずに1/Oポートのデータを読めるので、テスト が寒になります。

V Brill

h3170 h3070.20 h3170 20 h3 これは1/Oポートの70日を読み出したものです。00日 というデータが表示されています。これは、テキストウ ィンドウ (P. 255参照)の上位パイトのデータです。テキ ストウィンドウの上位パイトは〇コマンドで設定できる ので、ためしに20日 に設定します。

なお、テストが終ったなら〇コマンドで必ず00H に戻 しておいてください。それには、次のようにします。

h] 070, 00 [.]

■ L コマンド (Disassemble=ディスアセンブル)

入(力)形成 上 開始推圖,終了番地

例 [h] L0000, 001F [2] (0000H 番地から001FH 番地まで、逆アセンブルします)

例 2 h] L0000 ② (終了番地を省略した場合は開始番地から+ 15~17パイト分弟アセンブルします)

例3: h] L, 002F (開始番地を省略した場合は,逆アセンブルの終了 した番地から、指定した番地まで逆アセンブルし

ます)

Lコマンドは遊アセンブラで、マシン語のプログラムをインテル社の8080= ーモニックに責すものです。この逸アセンブラは、マシン語のプログラムをデ バックするのに健います。すなわち、マシン語のプログラムを遊アセンブルし てみて、もとのアセンブリ言語と同じであるかどうかを見るのです。もし、同 になたらなかったならば、ハンド・アセンブルが開途っていることになります。

▼解析にも使える

また、Lコマンドはマシン語プログラムの解析にも用いられます、解析するのに12撮製はわかりにくいですから、逆アセンブラにかけてアセンブリ言語に直してから解析を行います。ただし、Dコマンドを使用して文字列を除いてからでないとうまくいきません(インテル社の8080エーモニックだけを使用していればですが…)。

-(h31.888	88.86	81F		
	0680	F3		DI	
	8001	31	E1A0	LX1	SP.ELAB
	0004	C3	3BE5	JMP	3BE5
	0007	0.0		NOP	
	8000	7E		MOA	A.M
	8009	E3		XTHL.	
	BBGA	BE		CMP	3.5
	898B	23		ENX	H
	888C	E3		XTHL	
	800D	C2	0393	JNZ	0393
	8818	23		INX	H
	0011	7E		MOV	A . Et
	8812	FE	3A	CPI	3A
	0014	DØ		RNC	
	0015		0A15	JMP	0A15
	0018	F5		PUSH	PSV
	0019		ED42	CALL	ED42
	001C	C3	5925	JMP	5925
	801F	88		NOP	

THE PROPERTY.

N₈₀ - BASIC の ROM の 内 容 をディスアセンブルしたと 8 の例 です、 測り込み (P.241参照)を禁 让してから、29 ック・エリアを 決めて 3BE5H番地ヘジャンプし ていることがわかります。このよ うに、しつマンドは解析もできま す。

第1 初級者から中級者までのマシン語

また、Lコマンド実行中に実行を一時中断したいときなどは、次のようにします

画面の表示を一度停止させる………[CTRL] + [S] 再画面を表示させる…………[STOP]。[CTRL] + [C] 以外のキー

■Mコマンド (Move=ムーブ・メモリ)

例 h] MD000, D00F, D100 [2] (D000H 番地から D00FH 番地までのデータを D100H からにコピー)

Mコマンドは、指述された暴地のメモリのデータを指定する基地へ転送する ものです。このコマンドの健い方は3つあります。第1に、相対番地のプログ みを作ったときのテストに健えます。第2に、健用しているメモリのエリア を他のプログラムが健用する場合。エリアのメモリ内容を一時退避するのに使 います。第3に、マシン温のプログラムを実行するときに「備としてそのプロ グラムをとっておくのご機用できます。

.

- D000 88-90 20-21 00-02 00-03 20-04 20-85 02-26 02-07 00-08 00-09 00-0A 00-0B D00C 80-0C 00-0D 00-0E 80-0F 80-
- 00000 80 81 82 83 84 85 86 87 98 89 84 85 9C 8D 8E BE
- D000 80 80 82 80 02 80 08 08 12 80 80 80 02 82 02 88
- h)M0ed8.089F.D:08 h)DD189 D180 MM 01 02 03 04 85 05 07 08 89 8A 0B 0C 0D 8E 8F

▼画面

- ①Sコマンドを使用して、D000H 番地から D00FH 番地までに00H~0FH までのデータを書き込みます。
- ②Dコマンドで、データが書き込まれているかどうかを確かめます。
- ③これから転送する D100H~D10FH 番地までのデータを調べます。これで、 D000H~D00FH 番地までのデータとは違うことがわかります。
- ④Mコマンドを使用して、D000H~D00FH 番地までのデータを D100H 番地から転送します。

⑤Dコマンドを使ってデータがちゃんと転送されたか確かめます。転送されていることがわかります。

■ O コマンド (Output=アウトブット)

入力形式

- O ポートアドレス, テータ
- 例 ■] O 30, C0 □ (I/O ポート30H に COH というデータを送ります)

○コマンドは、指定したポートアドレスにデータを出力します。 非に自作した 1/0 ポートをテストするのに使います。直体 1/0 ポートにデークを出力できるので、テスト映画が坂くなり 未常に便利です、PC-8801mkIISR は 1/0 ボートを利用してマシンのシステムを作っていますので、このコマンドを使用してイタス方ができます。 システム全体がわかれば、イタズラではなくてシステムのコントロールができます。 たとえば、モノクロ画面からカラー画面に変えることなどもできます。

▼前面



①1/0 ボートの30H のデータが何であるか調べます。す ると、C3H というデータであることがわかります。 ②Oコマンドを使用して I/O ボートの30H に C0H とい

うデータを出力してみましょう。その結果とうなるでしょうか? 実行してみてください、多分、画面にカラーの "=" が表示される はずです。ここでは、画面のモードに関するコントロールをしていることがわ

かります.

③このままでは画面の文字が読みとれないので。○コマンドを使用して、元の アータを出力してあげます。すると、元のモノクロ画面になります。

■Pコマンド (Printer sw=プリンタ・スイッチ)

入力形式

P

例 h] P ☑

PコマンドはDコマンド、Lコマンド、CTRL-Dコマンドを実行したときに プリンタに出力するか、しないかを決定するものです。これは側面に出力され るものと同じものをプリンタに出力するものですが、人力したコマンドまでは プリンタに出力されませんので、それも出力したいときは COPY キーを使っ た方が良いかもしれません。

また、プリンタが ON のときと OFF のときではプロンプトが違いますので。 プリンタに出力されるかどうかすぐにわかります。プロンプトの違いは画1-8の

第1章 初級者から中級者までのマシン語

ようになります.

表1-8 プロンプトの違い

	プリンタ ON	プリンタ OFF				
プロンフト	h)	h)				
70271	q)	q]				

Pキーを1回押すとブリンタが ON に、もう1 座押すと OFF になります。

画面画

b)開

1回Pを押すと、h)が現れてプリンタがONになった ことがわかります。もう1度Pを押すと、h]が現れてOFF になったことがわかります.

■ Rコマンド (Read tape=リード・テープ)

入力形式 R ファイル名

例 I h] R test [(test というファイルをサーチしてみつけたら、ロードし ます)

例2 h] R. (最初に見つけたファイルをロードします)

注……ファイル名は6文字以内です。6文字を越える場合は、最初の6文字の みが有効となります。

Rコマンドは、カセットテーブからデータをロードするコマンドです。ファ イル名を指定しない場合は最初に見つけたファイルをロードし、ファイル名を 指定した場合は横定されたファイルを見つけるまでサーチします。

平 三 節

DIRTEST Found: TEST

hlR

①ファイル名を指定したときは、このように サーチしたファイル名を表示しロードしま 5. ②ファイル名を指定しなかったときは、 検初

に見つけたファイルをロードします。

■Sコマンド (Set memory=セット・メモリ)

入力形式

S開始和

例 h] SA000 [-]

Sコマンドは、メモリのデータを書き替えるコマンドです。データを変更し

ないでデータの確認をするときにも利用できて便利です。

Sコマンドを実行すると構定した悪地が表示され、次にそのときのデータが 表示されてデータの人力符ちとなります。ここでデークを実更する場合は、順 領データを入力しスペニスキーを押します、変更しないと多はスペニスキーを 押します。すると、データは前のままで次の番地へ進み、次の番地のデータを 表示して人力待ちになります。もし、Sコマンドを終了したい場合には、リタ ーンキーを押します。データをセットした後にリターンキーを押すと、最後の データはメモリにセットされます。

なお、数値データは16進数は最後の2桁、8進数は最後の3桁が有効です。

h1SD020

787-308 08-81 08-82 08-83 08-84 08-85 08-86 08-87 08-88 08-88 08-88 08-88 08-88 08-88

h100888 D888 88 81 82 83 84 85 86 87 88 89 8A 88 8C 8D 8E 8F

w ini m

A000H 番地から00H~0FH までのデータを入力し、Dコマンドで確認。 また、メモリの番地を1つ進めたり、戻したりといった細かい動作は次のようにします。

- ・メモリの暑地を1つ進める………スペース・キー
- ・メモリの番地を I つ戻す……………CTRL + B, ←
- メモリへ戻る…… RETURN

■ TM コマンド (Test Memory=テスト・メモリ)

入力形式 TM

例 h] TM2

このコマンドはあまり使用することのないものですが、定期的に使用することによって、マシンの高信頼性を得ることができます。TM コマンドを実行すると、本体に内蔵されている RAM がすべてチェックされます。

▼不幸にも BEEP 書が鳴ったら…

TM コマンドが実行されると、CRT 上にいろいろな文字が表示されますが、 気にする必要はありません。もし、不幸にも Beep 音が鳴り様かたら、CRT を 見てください、CRT 上に不良メモリのアドレスが表示されるはずです。こんな ときには、自分で検罪しようとはせず、Bit-INN などに連絡した方が良いでしょう。テストは、次の順序で行っています。

■ + 章 初級者から中級者までのマシン語

- ① Main RAM -------64KB
- (2) Graphics VRAM (Green)16KB (3) Graphics VRAM (Red)16KB
- (4) Graphics VRAM (Blue) ·······16KB
- (5) Text RAM32KB

テストにかかる時間は約12分です、無事テストが終了しますと。

Test Complete!

と表示されます。マシンをスタートさせるにはRESETボタンを押します。テストすることによって、システムで使用している RAM エリア内のデータが壊されてしまうので、コールド・スタートをかけなければならないのです。

■ V コマンド (Verify tape=ベリファイ・テープ)

入 力 形 式 V ファイル名

例! h] V ② (最初に見つけたファイルとメモリのデータを比較)

例2 h] V TEST ☑ (TEST というファイルをサーチして、見つけたらメモリのデータと比較します)

Vコマンドは、カセットテープのデータとメモリのデータを比較するコマンドです。カセットテープに無事データをセーブできたかどうかをチェックするときに使用します。

もし、エラーが発見された場合は"?"を表示しますので、もう一度セーブ し直す必要があります。



■Wコマンド (Write tape=ライト・テープ)

入 万 形 団 W ファイル名,セーブ 番地,セーブ終了番地

例 I: h]WTEST, D000, D0FF ☑ (TEST というファイルネームをつけた場合)

例 2 h] W. D000, D0FF (ファイルネームをつけなかった場合)

Wコマンドは、指定したエリアのメモリのデータをカセットテープにセーブ (書き込み)するコマンドです。ファイルネームはつけなくても可ですが、後の ことを考えるとつけた方が便利でしょう。

の画面

h]W.A000.A0FF h]WTEST.A000.A0FF ▼画面
①ファイルネームを指定しなかったときの画面
②ファイルネームを TEST と指定したとき

■ X コマンド (eXamine=イグザミン*レジスタ)

入力形式 X レジスタ名

例 (h] XA ((A レジスタが表示されます)

例 2 h] X (すべてのレジスタが表示されます)

XコマンドはCPUのレジスタを表示したり、変更したりするのに使用します。このコマンドは他のコマンドと組み合わせて。プログラムをデバッグするときに有効に働いてくれます。レジスタのデータを見ることができるので、現在のプログラム・カウンタのデータを読めば、どこのアドレスでストップしているががわかるからです。

これを利用してGコマンドで、ブレイク・ポイントを設定するとデバッグできます。

ここで、使用されているレジスタの記号を表1-8に示します。

■1-9 Xコマンドで使用 され■レジスタ酬記号

A, A*	アキュムレータ	8ビット
8, 81	BCペアレジスタ	16ビット
D, D°	DEベアレジスタ	16ピット
H, H1	HLペアレジスタ	16ピット
1X	IXインデックス・レジスタ	16ピット
IY	iYインデックス・レジスタ	16ピット
	レジスタ	8ピット
F, F*	フラグレジスタ	8ピット
PC	プログラム・カウンタ	16E 7 1
SP	スタック・ポインタ	16ピット
(注) 'が	付いているのは補助レジスタを示	によす.

第1章 初級者から中級者までのマシン語

B. D. Hはそれぞれ16ビット・ペアレジスタとして扱われていますので、 8ピット分のレジスタを変更するときは、変更しない残りの8ピット分のデー タも入力する必要があります

レジスタ名を入力しますと、そのレジスタのデータを表示して新しいデータ の入力待ちになります (これはSコマンドと同じ動作をします).

hix A :88 F :PZ---E-- 8 :8886 D :EDCC H :8881 A':88 F':PZ---E-- B':8888 D':FF7B H::8911 1X:2080 1Y:7DC9 1 :F3 PC:8000 5P:E5F9

▼画面 1

すべてのレジスタを表示させたときは、単行にわたって出力されます。

▼画面 2 hixa Aレジスタのデータを変更したときは、A A : 88=3E

A : 00 = R :0000= D : EDCC= H :8801= A':22= F : MZ --- 0--= 8::9000= D::FF7B= H::8342= IX:1206= IV: BRER=

PC:0000= SP:E5F9= レジスタのもとのデータも表示されます

VIII I

各レジスタのデータを変更しないようにス ベース・キーを押していったときは見やすく なります

■ HELP コマンド (HELP=ヘルプ)

入力形式 HELP about CTRL+A

例 I HELP (デキーは押す必要がありません)

例2 CTRL+A (***キーは押す必要がありません)

HELP コマンドは、モニタを使用しているときにモニタコマンドなどを忘れ た場合,使用するものです。[HELP] または [CTRL] +[A] を押すと、CRT に簡単 なモニタコマンドの種類とセットするバラメータが表示されます。

このヘルプ機能は、ちょっとモニタコマンドを忘れたり、マニュアルを引く のが面倒なときに使用できますが、ヘルプ機能でモニタコマンドをすべて理解 することはできません。

Monitor has following commands.

a <address> assemble source text lines.

bh or bq select radix (hexa decimal or octal).

d (start).(end) dump contents of memory.

e <start> change contents of memory by screen editor.
f <start>.<end>.<const> fill memory by the constant.

g (start).(breakl).(break2) execute user program.

(oprt) read input port.

1 <start>.<end> dis-assemble program in memory.

m (start:s).(end:s).(start:d) move contents of memory block(s=src.d=dest).

m (port).(new value) output new value to the port,

p toggle print switch.

r (file name) read cassette file. s (address) substitute memory contents.

tm test memory.
v <file name> verify cassette file.

w (file name).(start).(end) write cassette file

x or x (register name) dump all CPU registers or change the register.

CNTL-b return to BASIC.
CNTL-d (dr).(sur).(tr).(sec) dump contents of sectors.

[..<sec>] (sur) is optional.

CNTL-r (dr).(sur).(tr).(sec) read sectors into memory.
.(start).(end)

Ok O

▼ ■ 150

ヘルプ無比が表示された画面。ただしディスクバージョンでは、CRT 表示の 知めの部分がスクロールして出てきません。

■ CTRL-B コマンド (リターン)

入为形式 CTRL+B

例 h] CTRL+B

このコマンドを使用すると $N_{\rm sa}$ -BASIC モードに戻りますが、BASIC アログラムをこのままの状態で使用するときには注意が必要です。モニタでマシン語を操作するときには、初めに指定したマシン語エリアのみに使用してください。 $N_{\rm sa}$ -BASIC のシステムで使用しているメモリ・エリアを勝手に操作すると、

■ | 章 初級者から中級者までのマシン語

N₈₈-BASICモードに戻ったときに動作が保証されなくなってしまいます。 最 悪の場合は、禁走ということにもなりかねません。

■ CTRL-Dコマンド(ダンプ・ディスク)

入力形式

CTRL+D ドライブ番号, サーフェス圖号 (ティスケットの表記), 表示開始のトラック番号, 表示 のセクタ番号, 表示終了のトラック番号, 表示終了の セクタ番号

(注) トラック、セクタ番号は16進数でなければなりません。

例 | h] CTRL+D1, 0, 0, 1

(ドライブ1,サーフェス0.表示開始の トラック番号0,表示開始のセクタ番号 1で、表示開始のセクタのみ表示します)

例2 h] CTRL+DI, 0, 0, 1, 0, 13

(トラック番号が 0 でセクタの i ~l3まで表示します)

このコマンドは、フロッピイディスクのデータを CRT に表示させるもので す、データがディスクにセーブされたかどうが確かかたり、ディスクのどとに 書き込まれたのかを調べたりすることができます。さらに、他のディスクの解析にも使用できます。しかし、フロッピィディスクに関しての正しい知識を持っていないと、ディスケットを壊すことにもなりかれません。ただ、このコマ ンドはデータを見るだけなので、それほどの配する必要はよりません。

なお、表示終了トラック番号と表示終了セクタ番号を指定しないときは、表 示開始セクタ1セクタのみの表示になります。

bl^dl.8.0.1 Track 88.Surface.88.Sector 81 9888 18 81 83 21 68 C1 AF 32 84 EC 81 82 HE 11 11 MB 0818 3A 5D EF 87 28 64 86 82 1E 18 CD F4 C0 30 08 3A 8828 B4 EC FE 83 28 F4 C9 3A 61 EF 87 28 18 3A D7 79 8838 FF 34 38 BB DB 31 E6 88 28 83 CD 89 C1 3A 62 EF SE 67 CD D2 37 3E EF 8846 B7 CA 68 CI RR 37 AF CD D2 37 CD 47 38 2F 3E 81 3A B3 8B B7 CA 88 8868 32 MM 88 3A 5D EF 3E 89 8880 EF 8E 89 CD 80 C1 CD AE CI 18 78 89 28 D7 F5 3A 0000 3A 02 86 CI 88A8 83 88 FE 11 28 85 3E FF 32 FD C1 F1 FE 83 28 DB 88B8 3A 83 88 B7 88C8 81 32 SD EF 3E FF 34 39 F1 32 5D EF C9 80 88D8 88 88 86 82 80 96 88 86 88 88 88 88 88 88 88 88 eggs as en se an se at at at at at an an an an at at at at at 88F8 88 88 88 88 88 AF 3C E5 D5 C5 CD 9A 35 C1 D1 E1 C9

23\7> \49R23 /

東浦市

ドライブ1, サーフェス 0, トラック 0, セクタ 1 のときの画面で、IPL というプログラムが入っています。

■ CTRL-R コマンド (リード・ディスク)

入力 削減 CTRL+Rドライブ番号、サーフェス番号、リー ドニのトラック番号、リードニのセクタ番号、デ^{ミイフ} このカ場的アドレス、書き込み終了アドレスーメモリ

(9) CTRL+RI, I, 5, I, B000, B0FF

(ドライブ1, サーフェス1, リードトラック5H, リードセクタIH, リード 願始アドレス 8000H、リード終了アドレス 80FFH)

このコマンドは、ドライブ番外、サーフェス番外、トラック番外、セクタ番 分で指述されたセクタから、再き込み開始アドレス、軽子アドレスで指述され たメモリの番地ペテータをロードします、1セク与は128×バイトですが、これよ り扱いフログラムでも長さに応じてロードしてくれますので、リード開始のセ クタのみを指定すればよいのです。128×バイトより短くても、メモリのリード開 始アドレスと終了アドレスまでのバイト数しかロードされず、他のメモリ・エ リアは多単されませんのでかかです。

hlor.	1.1	.5.	1.D	398	DØ1	EF											
hillDi								-2									
DSBB	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	46	4F	0.0	ABCDEFGH1 JKLMNO
DBIR	88	00	818	88	010	ØB	SB	13 13	69	0.0	0.0	8.0	60	0.6	88	6.0	
D828	993	88	\$5.83	6.6	0.0	88	919	88	SIB	0.0	a a	03	8.8	88	84)	818	
0030	0.0	33	G 63	報日	98	813	36	98	89	9.0	88	3.0	00	66	89	6.0	
D8 48	9.0	80	98	10.03	9.9	80	90	00	88	88	90	20	8.8	99	88	0.0	
0050	88	93	88	68	313	9.0	88	9.8	00	88	0.0	8.0	6.6	88	00	UG	
DUGS	98	88	08	8.0	99	88	00	9.6	0.0	98	8.8	88	88	849	88	1310	
D678	1313	89	19 [3]	8.8	98	80	UB.	88	60	1313	80	8.0	45.8	66	88	86	
DBBB	98	88	86	88	88	88	8.69	00	88	99	8.0	98	88	13.63	BB	0.69	
0898	913	88	819	88	013	89	88	8.6	89	35	88	0.8	0.8	88	88	80	
DRAG	9.8	88	0.0	10 11	0.0	88	818	88	88	88	88	8.8	6.6	36	68	0.0	
DRRG	9.6	88	00	68	00	98	0.0	00	0.0	8.0	88	88	66	88	9.6	88	
DRCS	0.0	8.8	33	88	9.0	88	SH	BU	69	0.0	815	0.0	0.0	45 (5	88	0.0	
0808	8.0	00	(2-13	013	915	98	(10)	15-65	0.8	1515	0.0	15 63	66	4349	13.8	98	
DREB	8.6	88	818	88	9.9	9.6	5313	15 13	88	BB	53 EI	8.8	66	616	9.8	13.69	
DSE6	98	818	913	88	98	810	00	(34)	9.0	66	00	88	88	89	0.0	砂切	

★ 画 温

①ドライブ 1、サーフェス 1、トラック 5、セクタ 1 からリードして、D000H 番地から D0FF 番地までにロードしたときのものです。

②ロードされたかどうかDコマンドで確めたものです。

第1章 初級者から中級者までのマシン語

■ CTRL-W コマンド(ライト・ディスク)

入力形式

CTRL+W ドライブ番号, サーフェス画号, セー | ナリスク ブ開始トラック番号、セーブ開酬セクタ番号。 ヤーブ自治アドレス、ヤーブ終了アドレス → メモリ

例 CTRL+WI, I, 5, I, D000, D0FF

(ドライブ1、サーフェス1、セーブ開始トラック番号5、セーブ開始セク タ番号 I、セーブ開始アドレス DOOOH、セーブ終了アドレス DOFFH)

このコマンドは、フロッドィディスクへメモリのデータをセーブします カ セットテープと原理は同じですが、フロッピィディスク上に細かい番地が割り 当てられているために、細かい点まで指定してやらなければならない点がカセ ットテープと別なります.

1セクタは256バイト単位ですが、指定したメモリのエリアだけセーブします のでバイトの大きさを気にする必要はありません。256バイトより大きいもの は、自動的に数セクタを使用します。

```
hJ~wl.1.5.1.D000.D0FF -- 0
hJ^d1,1,5,1
Track 05. Surface 01. Sector 01
8898 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F BB ABCDEFGHIJKLMND
8010 88 89 88 88 80 80 98 80 80 80 90 30 80 80 80 80 80
8628 98 68 68 68 68 88 88 88 89 88 89 80 80 86 88 88 68 88
                 86 69 69 69 69 69 69 69 69 69 69
8838 88 88 88 88 88
                 28 28 28 28 28 28 28 29 36 26 28
2046 00 96 96 08
              98
              9.6
                 88 88 68 68 68 68 68 68 68 68 88 68
9950 98 98 98 98
6868 88 86 888
            88 88
                 60 60 60 60 60 60 60 60 66 66 66
                 88 86 86 86 88 88 88 88 88 88 88
8870 80 80 80 88 86
            80 68 66 66 66 66 66 66 66 66 66 66
00 66 66 6866
            89 89 88 86 86 86 86 86 88 88 88 88 88
8898 88 88 88
86 60 86 89 89 86 86 86 88 88 88 88 88
99C8 00 00 00
8058 80 88 89 90 90 80 88 88 88 88 80 98 80 88 88 88
80F6 96 96 86 86 86 80 80 60 60 80 80 86 86 60 98 88 88
```

V TO B

①ドライブにセーブしたときのものです。

⑦CTRL+Dコマンドを使用して、セーブされたかどうか確めたものです。

モニタコマンドの説明を一通りしてきましたが、中にはあまり使用しないコ マンドもありますので、必要によって覧えていけば上分でしょう。

アセンブリ言語を使うために

人間の言葉に近いアセンブリ言語

マシンが直接に命令を受けつけてくれるのは、0と1の組み合わせの2進数 だけです。2:識数はマシンには便利ですが、そのままの形で理解することは人 間にとって困難です。マシン語を2-進数から16連数に変換して理解しようとし たのですが、やはり16進数でも理解は展離でした。

人間が理解しやすい文字列の記号を定めてマシン請表現に使用すれば、数字を扱うよりは数段理解しやすくなります。そこで考え出されたのが「アセンブ り言語」です、アセンブリ言語は、コンピュータが実行できる1命令に対して 1つの記号が対応しているように作られており、しかもその記号はマシンの命令の動作を理解できるような文字列を使用しています。

たとえば、

01110110B 2 進数表現 ① 76H 16編数表現 ②

• HALT アセンブリ 表現---(3) (• 英語で、止めるという意味)

これらの命令の慰はすべて同じ命令で「マッン (CPU)の動きを会てストップせ よ」というものです。この3つの表現の中でどれが一番人間に理解しやすいか は一日酸終ですね。③の"HALT"は何かを停止させることだなと頭に浮ぶこ とでしょう。しかし、①や③の表現では何のことを言っているのかさっぱりわ かりません。アセンフリ言語はだいたいの命令の内容を大まがに知ることがで きるので鍛えるのに都合かれいのです。

このように人間が理解しやすいようにした文字列の記号のことを「ニーモニック」と呼んでいます。ニーモニックはマシンの命令を表しています。インテル社とサイログ社とでは、同じマシンの命令でもニーモニックが違います。たとえば "HALT" はサイログ形式ですが、インテル形式では "HLT"となっています。

CPUによるアセンブリ言語の違いと共通点

費から使用されてきた伝統的な套現を知ることは、いろいろな点でアセンブ リ言語習得に役立つものです。この伝統的な表現について説明しましょう。

■伝統的なデータ■動の■■

まず、図1-11のような構成のコンピュータを考えます。ここで、CPU を中心として外側から内側に向ってデータが転送されることを "Load" とよいます。

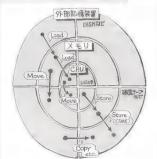


図1-11 コンピュータの構成と伝統的なデータ移動偏現

遊に、CPUから外側に向ってデータが転送されることを"Store"と言います。 レジスタAからレジスタBへの転送のように、同じレベルでの転送は、"Move" と言います。また、外部記憶装製間のデータ転送。"Copy"などの表現を用い ます。それは、外部記憶装製間では直接データ転送を行えず、コンピュータ本 体を使用して中継しなければならないからです。

■ 觀しみインテル形式、すっきりザイログ形式

インテル形式のアセンブリ 言語は伝統的な表現を数多く反映させていますの で、昔からコンピュータに馴れ親しんだ人には覚えやすく、サイログ形式のア センブリ言語より使いやすいと思われます。それに対してザイログ形式は、Z-80の命令セットが大き、複雑なために"Move"、"Store"などの表現を特で、 すべて"Load"一言で済ませており、数の少ないすっきりとしたものになって います。といっても、全面的に伝統的な表現を捨て去ったわけではありません が…。

このようにアセンブリ言語は各CPUによって違いがありますが、共通の部分も多いのです。この動どころを押えると、アセンブリ言語を早く習得できると思います。

アセンブリ言語表現の約束事

キーボードにある文字であれば、どんな文字でもアセンブリ言語表現に催え そうですが、アセンブラ (後で展明します)を使用したりするために、彼よる 父言は限られています。しから、記号としての文字列とデータとしての文字列 とに分けて考えておく必要があります。一般的に使用できる文字は次のような ものです。これ以外の文字を使うことはできませんので、しっかりと覚えてお いてください。

■使用できる文字

- ① A~Zの大文字(場合によっては小文字も使用できる)
- ② 0~9の数字
- ③ #, a, ?, -, などの記号も使えるものもある

こう見ると、アセンブリ言語で使用できる文字は、BASIC などに比べて非常 に限定されていることがわかります。わかりやすくするために使用する文字の 権難を少なくしているのだと思われます(もちろん、他にもいろいろ理由はあ りますが……)。

プログラムを読みやすくするためには、記号としての文字列の構成に注意することが大切になってきます。文字列を作るとき、英語の観りにしたりローマ 字の観りにしたりしますが、コメントを含めてどちらかに統一した方が良いで しょう、特にコメントは小文字が使用できたらその方が便利になります。文字 列とデータ列の約束率は一般的に次のようになっています。

■文字列とデータ列との約束事

■文字列------①初めの文字は必ずアルファベット
②6 文字以内

第 | 翻 初級者から中級者までのマシン語

正しい例	誤った例					
ABCDEF	1ABCDE					
LOOP	ABC DE					
LOOP3	L_ABC3					

● ■データ列…①初めの文字は0~9』での数字。もし A~F が使用された ときは 0 を確につける

(2) 2 准数: 8 准数. 16 進数の区別をはっきりつける

■ 0 がついたら数値データ

文字列の第1次字はアルファベットから前まっていなければなりません。これは、文字列を数所・ク列とをはっきり区別するためです。これは数似アー列にも言えることで、16進数を扱り場合に AーFまでのアルファベットを使用していますので、数似データの第1次字に AーFがくることもあります。そこで、文字列と区別するために、第1次字が A一Fになったときには、その前に0をつけて数似テータであることを表示します。

また、文字列などは予約記号 (予約語) と呼ばれるものもあります。予約記 号の中には CPU のレジスタ名も含まれますので、"BC"、"HL" などの文字列 は使ってはなりません。スペースも記号と記号の区切りとして傾用しています ので、使用することはできません。

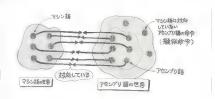
マシンにない。似命令

アセンブリ言語の命令はマシンの命令と1対1で対応しているわけですから、当然、アセンブリ言語の命令はマシンの命令数を含んでいるのですが、その他にマシンにはない命令をいくつか持っています。

マシンにはない命令を軽似命令と言います。軽似命令はアセンブリ言語で青いたプログラムをわかりやすくするのと、アセンブラ (後で説明します) に連絡するために用いられます。

アセンブリ言語の世界は図1-12のようになっています。

図1-12 アセンブリ言語の世界



■基本的な個似命令

基本な擬似命令には、表1-10のようなものがあります。

売1-10 基本的な擬似命令

擬似命令	由来語	意 味								
ORG	Origin	マシン語のスタート・アドレス (本来はロケーションカウンタ)								
END	End	アセンブリ側の終了								
DB, DEFB	Define Byte	指定された番地に1バイト・データをセット								
DW, DEFW	Define Word	指定された番地に2バイト・データをセット								
DC	Define Character	DBと同じ								
DC, CONST	Define Constant	指定した番地に定数をセット								
DS	Define Storage	指定された番地から指定されたバイト数を確保								
EQU	Equate	文字列に数値データを与える								
DATA	Data	DBと同じ								

ここに出てきた擬似命令を知っていないと、実用的なプログラムをアセンブ

■1章 初級者から中級者までのマシン能

り言語で書けませんので必ず覚えておきましょう。

アセンブリ言語のかたち

アセンブリ言語のプログラムの書き方は昔からあまり変っておらず、全く共 通しているといえるくらい等られています

命令は1行(1ライン)にひとつだけ書き。原則としてマシンの命令に対応 するように書いていきます。この1行に書かれた命令を文と呼んでいます。

BASIC のようにマルチ・ステートメントはできません。次に文の例を示しますので見てください。

a b c d

■ラベルは名札.

ラベルは命令などにつけられる名札で、BASIC の行番号に相当するものですが、必要のないところにはつけません。ジャンプ命令やコール命令などのときに参照する機識となるものです。

■ニーモニック, オペランド

実行命令で、マシンの命令を2つに分けて表しています。ニーモニックはしいて言えば大まかな命令で、オペランドは具体的な命令の操作を表しています。

■コメント

コメントは全くの覚え書き用でアセンブラは無視しますので、プログラマが 自由に利用できます。

では、具体的な例を示します。

ラベル ここを オペランド コメント

START: EQU 0A000H ; Start Address (START という変数に、A000Hをセット する)

ORG START (マシン語のプログラム開始番地はA000

Hとなる)

LD B, 03H ; Move B→03 (Bレジスタに03Hをロードせよ)

LOOP : LD A. 41H : Move A→41 (Aレジスタに41Hをロードせよ)

DEC : Decrement B-1 (Bレジスタから1を引く)

JP NZ, LOOP ; Jump Loop (ループというラベルのあるところへゼ ロフラグが立ってなければジャンプサよ)

END

このようになりますが、ニーモニックの"LD"はデータを移動せよという命令だけで、具体的にどうせよとは表現していません。どこに、どのデータを移動せまと表現しているのかオペランドです。しかも、オペランドは「、」を中心にして右から左へ移動するように書き表すことになっていますから、すっきりしています。

ハンド・アセンブルでマシン語マスター

アセンブリ 高語で表いたプログラムで、 すぐにマシンを動かすことはできません。 つまり、マシ idそのままの形ではアセンブリ if 前を理解できないのです。 マシムは2 連載しか受けつけてくれませんから、アセンブリ if 前で if ボンブログラムを何らかの方法で2 進数に変換してやらなければなりません。このアセンブリ if 指を 2 変化に変換することをアセンブル ff 業とは、2 つの方法があります。

- a. 手作業で行う方法 (ハンド・アセンブル)
- b. セルフ・アセンブラ(アセンブラとも言う)プログラムを使用する方法 これを図で表すと図1-13になります。

ハンド・アセンブルとは、人間がマシン語の対応表を見ながら、1命令ずつ アセンブリ言語からマシン語に実換していく作業のことです。たいへん面倒な 作業なのですが、マシン語を覆い始めた人は必ずこの作業をやらなければなり

第 | 章 初級者から中級者までのマシン語

ません、この作業に慣れるに従って、アセンブリ言語に対応するマシン語を覚 えていきますから、非常に好都合なものです。

ハンド・アセンブルをするのには、ちょっとした道具が必要です。それは、 アセンブリ言語とマシン語との対応表です。この本の付縁として対応表をつけ ましたので、ハンド・アセンブルの際に使ってください。

ハンド・アセンブルは、短いものなら短時間で確実にマシン語に変換できますが、プログラムが長くなるにしたがって、間違えずにマシン語に変換するのが難しくなってきます。その点、アセンブラを使えば、正しくアセンブリ言語で書かれていれば、間違いなくマシン語に変換してくれます。このプログラムを使用すると、面倒でバタを作りやすい作業をしなくて済みます。

この章では、マシン語をマスターするための基礎的な知識について説明しま した(フローチャートについては説明していませんが、興解されているものと して省いてあります)、ここまでのことがある程度わかっていると、決業からの ことがよく理解できると思います。



モノクロ画面から カラー画面への招待

キャラクタ単位で色をつける

さて、マシン諸についてひととおりの 知識を獲得したところで、この章からは 未書のメインである画面操作の説明に入 っていきます。 適価に自分の好きな色をつけられたら 流しいですね。まず、どうやって画面に 色を出しているのかを知らないと述にな

> よう、 Pトリビュートエリアを利用して、 テキスト画面上の文字を反転させたり点 減させたりします。 そして、 キャ・ソヤ 単位で色をつけたり、 バックグランドに 色をつけたりといったテキスト画画のカ

CRTのカラーの原理

CRTの基本は3原色

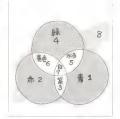
家庭にある TV はほとんどカラー TV です。 コンビェータ用の TV (CRT と も言います) 6 家庭用 TV を利用したものなので原理は同じです。 TV がカラー になるのは光の3 原色を利用したもので、 青。 赤。 縁の組み合わせによってで す。 TV のスクリーンの近くには青。赤、縁の機を整料が整ってあり、この3 原 色の光り方で色かつくことになります。 図2-1のようになりますか。これでは無 を含めて8 色しか出ません。しかし、家庭用の TV は無限色に近い色を出すこ とができます。 家庭用 TV では、青。赤。 縁の強弱を変えることによって様々 な色を作り出すのです。

■かけあわせれば色いろいろ

カラー CRT では、青、赤、緑の強 弱を変えることができません。それ で色は II 色のみになって しまいま す。しかし、 工夫次第では中間色も 可能です。 後輩のグラフィックス画 価の変で説明しましょう。

■ | 一からわかるように、縁と書を 発光させれば水色になり、縁と赤を 発光させると黄色になります。また、 青、縁、赤を発光させると白になり ます、キャラクタ単位で色をつける とき、この原理を知っていないと、 希望の色を出すことができません。

図2-1 ||原色による色



Section 2 テキスト画面の VRAMを知る

ディスプレイエリア(表示用)とアトリビュートエリア(装飾用)

CRT 画面に文字を表示させるためには、VRAM を操作する必要がありま す。この VRAM は、CRT の画面にメモリの番地を割り当てたものです。した がって、VRAM にデータを書き込んだり読み出したりすることが自由にでき 主士

VRAM の番地と CRT の画面上の位置関係は図■-2のようになっています。 このように VRAM は画面表示用(ここではディスプレイエリアと呼ぶことに 1.ます)と表示には直接タッチしない画面装飾用(アトリビュートエリア)と に分けられます。ディスプレイエリアは CRT の画面と一致しており、この番地 にキャラクタ・コードを書き込めば画面上に表示されます。アトリピュートエ リアは、表示されている文字をリバース(反転)したり、プリンク(点滅)し たり、あるいはシークレット (表示させない) したり、アンダラインとアッパ ラインを引いたりすることに使用するものです。また。カラーモードのときは 文字に色をつけるためにも使用されます。

アトリビュートエリアの使い方はプログラムを作りながら説明していきます が、ディスプレイエリアの番地は付録を利用してください。

F487 F43F (表示されない部分で、表示されているキャ ラクタに変化をつける時に使用される機能) アトリピュートエリア 西面装飾用 F418 FEED FF58 F490 F417 F48F FEDF FF57 (CRTの画面の位置と一致している) ディスプレイエリア 画面表示用一 図2-2 VRAMの書画とCRT画面上の位置画係 FE90 F3C8 F440 FF08

面表示に必要な キャラクタ・コ

カナやアルファベットも数値で扱う

コンピュータはアルファベットやカナ文字などもすべて教領として扱ってい ます。したがって文字などは対応する数値を用いなければ、希望の文字が表示 されません。その決められた数値をキャラクタ・コードと言います。たとえば Aという文字を表示させるには 41日を使用しなければなりません。パソコンで 使用されているキャラクタ・コードは ASCII (アスキー:米国の規格) コード に連擦しています。準拠しているといっても、同じコードなのは半分以下です。 たお、大型コンピュータでは ASCII コードは使用していませんので注意してく たさい.

PC-8801mkHSR (PC-8801, PC-8001) のキャラクタ・コードは *2-1のよう になっています.

1-85 1-85	0	1	2	3	4	5	6	7	8	9	Α	В	C	D	Е	F
0		D_E		0	(0	P		р	_	-			9	i		×
1	$S_{\rm H}$	D_1	1	1	Α	Q	а	q	_	~	0	7	+	4		FI
2	$s_{\rm X}$	D_2	n.	2	В	R	b	r	1000	4	г	1	7	X		华
3	${\rm E}_{\rm X}$	D_3	#	3	С	S	С	s		ŀ	J	ゥ	Ť	ŧ		Л
4	ET	D_4	S	4	D	Т	ď	t				I.	1	to	4	H
5	EQ	$N_{\rm K}$	%	5	Е	U	е	u		-		才	+	2	•	時
6	$A_{\rm K}$	s_N	&	6	F	V	f	V		1	7	カ	ň	3	4	5)
7	BL	EB		7	G	W	g	W		-	7	キ	×	7		秒
8	BS	c_{N}	(8	Н	X	h	х	1	Г	1	2	ネ	1)		
9	H_{T}	E_{M})	9	I	Y	i	У		7	2	5	1	n	¥	
Α	LF	SB	*		J	Z	j	Z		L	3.	2	^	Þ	•	
В	H_{M}	EC	+		K		k	1			*	++	Ł	D	+	

M] m
N ^ n
O o

7

アトリビュートエリアの 属性を知る

属性を変化させる

海川工で支付してこう

ディスプレイエリアにキャラクタを扱ぶさせるには、ディスプレイエリアの 霧塊にキャラクタ・コードを書き込みます。しかし、キャラクタを反転(リパ ース)させたり点滅(ブリンク)させたりすることはできません。そこでキャ ラクタ・コードとは直接関係ない情報をアトリビュートエリアに書き込んで反 転や広滅をさせます。

■属性変更は 20 回まで

アトリビュートエリアは1行につき40パイトで構成されており、1回の属性 を変化させるたびに2パイト使用されます。したがって、属性は1行について 20回までしか変えることができません。

アトリビュートエリアの構成を1行目を例にとって考えてみましょう。**図2**~ 『のようになります。どの行も番地が譲う以外すべて同じです。

初めの1パイトで何行目から属性を変化させるかを指定し、次の1パイトで どのように変化させるかを指定します。もし、1行全部を反転させるのであれ ば、最初の2パイトだけ指定すればその行全部が反転します。

■属性の指定方法

次に調性の指定方法について考えてみましょう。 脳性のデータはモノクロモードとカラーモードでは違ってきます。 カラーモードの場合は、反転や点減な どのほかにカラー指定しなければなりませんので、モノクロモードの属性指定 よりは機嫌になります。 属性コードの構成は個2-4のようになります。

図2-4 属性コードの構成



たとえばモノクロモードで1行を反転させたいときは、属性の番地(たとえば1行目なら F419H)に04H (00000100B) を書き込めばよいことになります。この属性の扱い方は次の項で練習してみましょう。

使い方あれこれ

この職性の利用のし方はたくさんあります。たとえば、リバースはメニュー プログラムを選択するときに、選択したプログラムを反転させるのに使用でき ます。プリンクはオペレークに注意を使すときに使用できます。またシークレ ットは、キーワードやパスワードのように機能性を必要とするものを表示させ ないために使うことができます。他にいろいろなプログラムで、アトリビュー トエリアの機性は使用できるので利用してください。

モノクロモードで アトリビュートエリアを操作する

文字 "A"の属性を変化させる



アトリビュートエリアの属性がわかったところで、その属性を他う解判をしてみましょう。最初の行に「A」という文字を表示させてから、リバース、ブリンク、シークレットに属性を変えて、最後にこの3つの属性を退化させてみようというわけです。 練習の順番は次のようにしてみましょう

練習 リバースのプログラム

練習3 シークレットのプログラム 練習4) 3つの混在するプログラム

文字をリバース(反転)させる

まず、画面の設計から考えていきましょう、 平順は、F3C8H 番地から「A」 を80個表がした後に、アトリビュートエリアの F418H 番地に 00日 をセットし、 F419H 番地にリバースをせるデータ 04日 をセットし、モニタへ反ればいいこ とになります (**822-6**, P78)。

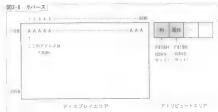
■・用するアドレスとデータ

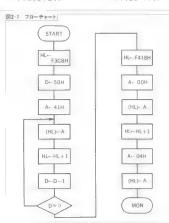
では、ここで使用するアドレスやデータをまとめておきましょう。

○プログラムのスタート・アドレス B900H

○ CRT のスタート・アドレス F3C8H ○アトリビュートエリアの桁数アドレス F418H

○アトリビュートエリアの属性アドレス F419H





41H

Oモニタのアドレス

E826H 04H

○属性データ

891A

○書き込みキャラクタ「AI

ここで騒性データが04日になっているのは、3 ビット目がしになればリバー スすることになるので、他はすべて0 になるようにするためです。属性コード みもう一度はてください

■フローチャート■プログラム

次にフローチャートを作成します。フローチャートを図1-7に示します。

プログラムリストはリスト2-1のようになります。プログラムの説明をしましょう。まず、HL ペアレジスクに CRT のスタート・アドレスをセットし、Dレジスクに80回ループを繰り返す50Hをセットします。Aレジスクに表示するキャラクタ・コードの4IH「A」をセットしたら、HL ペアレジスクが指定するVRAMのディスプレイエリアにロードし、それを50H 繰り返させます。

ループが終ったならばアトリビュートエリアのアドレスを HL ペアレジス タにセットし直します。このプログラムではセットしなくても HL ペアレジス タのデータは F418H (1行目のアトリビュートエリアの番地) になっています

●リスト2-1 Aをリバースさせる

END

	********	Monochrome Text	Mode *******
B988 E826 F3C8 F418 0804 8041	START: EQU MON: EQU CRTAD: EQU SKETA: EQU AV: EQU CHAR: EQU	GB986H GE826H GF3C8H GF418H G4H	:START ADDRESS :MONITOR ADDRESS :V-RAM START ADDRESS :V-RAM AV ADDRESS :REVERSE DATA #4H :CHAR DATA *A
5980 21CBF3 8903 1650 8905 3E41 8907 77 8908 23	UD LOOPI: LD LOOPI: LD INC	HL.CRTAD D.50H A.CHAR (HL).A HL	;! LINE DATA 80KETA :DISPLAY CHAR CODE
8989 15 896A C287B9 898D 2118F4 8918 3E80 8912 77 8913 23 8914 3E84 8916 77 8917 C326E8	TD TD TD TD TD TD TD TD TD TD	NZ.LODP1 HL.SKETA A.00H (HL).A	:DISPLAY NO START KETA :V-RAM AV DATA :MONITOR

が、いつもルーブが終ったときにアトリビュートエリアのアドレスになっている とは限りませんので、セットし直した方がベターです。それから F418H 番地に 0桁目から表示するデータの00H をセットし、F419H 番地にリバースするため の個性データをセットし、モニタに乗ります。

■実行するには

実行するときは、EコマンドやSコマンドを使用して B900H 番地から入力 して実行させてください。写真のように1行にAが表示されて反転している両 面になるはずです





■プログラムをディスクにセーブするには

なお、このプログラムをディスクにセーブするには、BASIC モードに戻って BSAVE コマンドを使用します。

BSAVE "mono 1. B", & HB900, & H 19 H [-]

で、プログラム2-1がセーブされます。「DUAD-88」を使用している場合はホット・スタート (STOP+リセット・ボタン)して BSAVE してください。

リバースに使用したマシン語命令

7 - 7 (40 (50/1) 0/5 + 7 7 HK HP 1

■ LD (ロード) 命令

・番多く使用される命令で、8ビットや16ビットのデータを CPU のレジス タ同士やメモリの間で移動させる命令です。データの移動といっても移動先の データは変化しませんから、データのコピーと覚えた方が良いでしょう。

LD r. n

(注: rは汎用レジスタ名を表し、nはまピット・データを表します)

書式 LD r, n

§5 モノクロモードでアトリビュートエリアを操作する

機能 8ビット・データを「で指定されるレジスタにロードします」

例 LD A, 41H

Aレジスタに4IHをロードします。

LD (HL), r

書式 LD (HL), r

機能 rレジスタのデータを HL ペアレジスタで示されるメモリの番地 にロードします。

例 LD (HL), A

HL ペアレジスタの内容を F3C8H とすると



A レジスタのデータ(41H)が F3C8H 番地へロードされます。

LD pr, nn (注:prはペアレジスタ HL, DE, BC を表します)

書式 LD pr. nn

機能 指定するペアレジスタに2バイトのデータをロードします。

例 LD HL, 0F418H

Hレジスタへ F4H (上位パイト) を、Lレジスタに 18H (下位パイト) をロードします。

■ INC(インクリメント)命令と DEC(デクリメント)命令

INC, DEC 命令は説用レジスタに1をプラスしたりマイナスしたりするもので、同数を数えたりするのに多く使用されます。また、DEC 命令は、次に説明するJP命令をセットして使用することが多くなります。この命令はペアレジスタでも使用できます。

INC pr

書式 INC pr

機能 指定されたペアレジスタに 1 を加算し、その結果をペアレジスタ に入れます (ペアレジスタ ← ペアレジスタ+1).

F3C9

例 INC HL

実行前の HL ペアレジスタの内容 F3C8

実行後の HL ペアレジスタの内容

実行の結果、HLペアレジスタに上が加算されました。

DEC r

書式 DEC r

機能 指定されたレジスタから1を■■し,結果をレジスタに入れます。 その際、フラグも変化します(レジスタ ← レジスタ+1)。

DEC D

□レジスタから1を減算します。その結果、たとえばDレジスタ の値が00HになったときはZ(ゼロ)フラグは1になります。

■ JP (ジャンプ) 命令

JP 命令はプログラムの流れを変えるために使用されます。フラグの内容によって流れも変化するものもあります。

JP nn

書式 JP ジャンプ先の番地またはラベル

機能 nn で指定された番地に無条件でジャンプ(プログラム・カウンタ にnn 値をロード)します。フラグの影響は量けません。

(9) JP 0A000 H

メモリの A000H 番地へジャンプします.

JP NZ, nn

書式 JP NZ, ジャンプする番地

機能 ゼロフラグが D のときに指定する番地にジャンプします。ゼロフラグが L のとき (レジスタの値が D) に次の命令を実行します

1991 DEC B

JP NZ, 9000H

DEC A ゼロフラグが 0 のときは 9000H 番地へジャンプし、ゼロフラグが 1 のときは "DEC A" の命令を家行します。

この「JP Z, nn」命令は使用していませんが、「JP NZ, nn」とは反対の命令ですから、対にして覚えましょう。

JP Z. nn

書式 JP Z, ジャンプする番地

機能 ゼロフラグが 0 のときに次の命令を実行します。ゼロフラグが 1 のと II (レジスタの値が 0) は指定する番地にジャンプします。

例 DEC B

JP Z, 9000H

DEC A

ゼロフラグが 0 のときは "DEC A" の命令を実行し、ゼロフラグが 1 のときは 9000H 番地へジャンプします。

プログラムリストの見方について

リスト2-1のようなプログラムがこれから先たくさん出てきますので、このプログラムリストの見方を説明しておきましょう。プログラムリストに説明を加えると、図2-8のようになります。

■ EQU 命令を使うわけ

EQU 命令が多く使用されていますが、次のような理由によるものです。プログラム中の次の箇所に注目してください。

CHAR : EQU 41 H

LD A, CHAR (41H) -- ①

①のLD命令では、Aレジスタにキャラクタの"A"で41Hを直接に入力しないで、EQU命令を使って"CHAR"に41Hを入れてから、"CHAR"という ラベルでAレジスタにロードしています。一見複雑なようですが、この方が便 別かのです。

というのは、今回のプログラムでは1回しか"A"レジスタにキャラクタ・ コードをロードしていませんが、2箇所以上でてくる場合は、キャラクタが実 わるたびにプログラムを書き値さなければなりません。2箇所ある場合は、2 箇所を別のキャラクタ・コードにしなければなりません。

しかし、ラベルで定数(この場合はキャラクタ・コード)を宣信しておき、 LD命合などを用いてラベル名でキャラクタ・コードをロードできるように記述しておけば、何箇所あっても EQU 命令の定数を一箇所書き直すだけで良い のです、後のことはアセンブラがすべてラベルを定数に置き換えてくれます。

このように EQU 命令を使用することによって、時間の短縮とミス防止をすることができます。つまり、本当のマシン語 (擬似命令でない命令) の部分をいじらないので、プログラムがおかしくなることもないのです。

■まずはハンド・アセンブルから

アセンブラを使用すると、ハンド・アセンブルでは考えられないほどスピー ドが上がりますが、初めての人は必ずハンド・アセンブルを行ってください。 その方が、遠回りのようでも近道となります。

1行ブリンク(点滅)させるには

ブリンクさせるといっても特別なことをしているわけではなく、アトリビュ

			### ### ### #### #####################	, A 828H) K K
200 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		ORG ILLE	7/	
が入れ器ですいる 一点はパイトと下位パイト	(2) (2) (2) (3) (3) (4) (4) (5) (5) (5) (5) (5) (5) (5) (5) (5) (5	8988 6983 6987 8987 8988	8989 8984 8918 8918 8913 8913	

ートエリアの属性を奏えているだけです。リスト2-2を見てください、リスト2 -1のプログラムと違うのは、属性データだけです。つまり、属性データ以外は リスト2-1のプログラムと全く同じで良いということです。B900日 番地から実 行させると、1 行ブリンク (点論) するはずです。

●リスト2-2 Aをプリンクさせる

				Text Mo	de «manenane»
	: レンシュウ				
B900		EQU	явавян		:START ADDRESS
E826		EQU	0E826H		: MONITOR ADDRESS
F3C8	CRTAD:		DF3C8H		: V-RAM START ADDRESS
F418		EQU	9E418H		: V-RAM AV ADDRESS
0002		EQU	026		: BLINK DATA 82H
8941	CHAR:	EQU	41H		CHAR DATA 'A'
		ORG	START -	こだけがリス	12-12連う
	1				
B900 21C8F3		LD	HL, CRTAD		
8903 1650		LD	D.50H		: 1 LINE DATA 80KETA
8905 3E41		LD	A, CHAR		:DISPLAY CHAR CODE
B907 77		LD	(HL1,A		
B908 23		INC	HL		
B909 15		DEC	D		
B90A C207B9 B90D 2118F4		JP LD	NZ.LOOPI		
B910 3E00		LD	A.86H		DISPLAY NO START KET
B912 77		LD	(HL).A		DISPLAT NU START KET
B913 23		INC	HL.		
B914 3E82		LD	A.AV		: V-RAM AV DATA
B916 77		LD	(HL).A		A-KHM MA DWIM
B917 C326E8		JP	MON		MONITOR
032000	:	ot.	(HOIN		- WORLESSE
B91A		END			

1行を消してしまう~シークレット

このプログラムもリスト2-2と同様, 注の部分の属性データが変わっているだけです。リストを示しましょう(リスト2-3)。

実行させると1行が消えます。しかし、ディスプレイエリアのF3C8Hから F417H 番地までは、"A"のキャラクタ・コードである41Hが開遠いなく書き 込まれているはずです。モニタのEコマンドで確認してみてください。

■ 落し穴や異次元の入口にも使える

ディスプレイエリアにデータが残っていると、使い方によっては前白いこと ができます。たとえば、バスワードの確認のために使用できますし、ゲームな どでは落し次や異次元の人口のように、普段は消しておいて近くに水たり触れ たりしたときに現れるようにすれば、変化のある動画をつくることができるで 1 6 3

■リスト2-3 1行をシークレットす■|

	************************************	Monochrome	Text Mode	· 公司资格款条款价格的
3908 1826 73C8 F418 3801 8941	START: EQU MON: EQU CRTAD: EQU SKETA: EQU AV: EQU CHAR: EQU	88988H 8E826H 8F3C8H 8F418H 8F418H	こだけが違う	START ADDRESS MONITOR ADDRESS V-RAM START ADDRESS V-RAM AV ADDRESS SECRET DATA BIH CHAR DATA 'A'
3988 21C8F3 3983 1658 3985 3E41 3987 77 8988 23 8989 15 8988 C287B9	LD L	HL.CRTAD D.50H A.CHAR (HL).A	- 1.11 (7 地)	: LINE DATA 88KETA :DISPLAY CHAR CODE
B90D 2118F4 B910 3E00 B912 77 B913 23	LD LD LD 1NG			;DISPLAY NO START KET
8914 3E01 B916 77 B917 C326E8	LD LD JP	A.AV (HL).A MON		: MONITOR
B91A	EN)		

1行にいろいろな属性をもたせる

ここでは、5種類ある属性をすべて使用してみましょう.

■反転文字を点滅―リバース+ブリンク

属性は組み合わせて使用することができますので、リバースとブリンクを組み合わせてみましょう。リバースとブリンクを組み合わせたプログラムをリスト2-4に示しますが、このプログラムもリスト2-1とほとんど同じです。

■リスト2-4 リバース+ブリンク■

	:***** Monochrom	e Text Mode *******
E826 F3C8 F418 9896	LUSSAT / 4-1 START: EQU 88988H MON: EQU 8E226H CRTAD: EQU 8F3C8H SKETA: EQU 8F418H AV: EQU 8F418H CHAR: EQU 41H EQU 4TH EQU 4TH EQU 5TART	:START ADDRESS :MONITUR ADDRESS :V-RAM START ADDRESS :V-RAM AN ADDRESS :KEVERSE AND BLINK DATA 86F

B988 21C8F3 B983 1658 B985 3E41 B987 77 B988 23	L00P1:	LD LD LD LD	HL.CRTAD D.50H A.CHAR (HL).A	:1 LINE DATA 80KETA :DISPLAY CHAR CODE
8909 15 890A C20789 890D 2118F4 8910 3E00 8912 77		DEC JP LD LD	D NZ.LOOPI HL.SKETA A.00H (HL).A	:DISPLAY NO START KETA
B913 23 B914 3E66 B916 77 B917 C326E8		INC LD LD JP	HL A.AV (HL).A MON	:V-RAM AV DATA :MONITOR
BSIA		END		

この属性データは3ビット目(リバースのビット)と2ビット目(ブリンクのビット)を1にするだけで、他のビットはすべて0にします。実行させると 反転文字になって点滅を繰り返します。

■文字の上下に線をひく一アッパライン+アンダライン

BASIC などではサポートされていないアッパラインとアンダラインを表示 させてみましょう。リスト2-5がそのプログラムです。

前のプログラムとの違いは、属性データを 30H にするだけです。実行させる と、文字の上と下に線が引かれます。

■リスト2-5 アッパライン+アンダライン ■

	*****			Text Mode *******	
			-2 (UPPER	8 UNDER LINE:	
B980	START:	EQU	6B986H	:START ADDRESS	
E826	MON:	EQU	8E826H	:MONITOR ADDRESS	
F3C8	CRTAD:	EQU	0F3C8H	:V-RAM START ADDRESS	
F418	SKETA:	EQU	0F418H	:V-RAM AV ADDRESS	
0030	AV:	EQU	30H	: UPPER AND UNDER LINE DATA 0	6
0041	CHAR:	EQU	41H	:CHAR DATA 'A'	
				ここだけが進う	
		ORG	START	12 17 O'18: 7	
B900 21C8F3		L.D	HL, CRTAD		
8903 1650		LD	0.58H	: 1 LINE DATA 88KETA	
B905 3E41		LD	A, CHAR	:DISPLAY CHAR CODE	
8987 77	L00P1:	LD	(HL),A		
B998 23		INC	HL		
B989 15		DEC	D		
B90A C207B9		JP	NZ.LOOP1		

§5 モノクロモードでアトリビュートエリアを操作する

■8種類の属性を | 行に表示

属性データを変えて8種類の違ったものを表示させましょう。8種類の組み 合わせにしたのは、シークレットと他のものを組み合わせてもラインが表示さ れているかの確認がとれませんので意味がないからです。意味があるものだけ を確よと8種類になるわけです。

属性の種類が8種類ありますので、1つの属性に対して10文字分割り当てる ことができます。画面に図2-9のように表示させることにしましょう。

図2-9 画面標準



これに対応する属性コードは■2-2(P.90)のようになります。

アトリビュートエリアは2パイトで1回総社を変化させることができます。 今回は8回総性を変えるわけですから、16パイト使用することになります。初めのパイトに基础する指の使服を書き込んで、次のパイトに機性データを書き込んでやることを8回繰り返せば良いことになります。プログラムでは、1パイト書き込むたびに IL ペアレジスタに 1 を加えて次の季慮にしていくことによす。基本的なことは、リスト2-1のプログラムと変わりありません。

プログラムはリスト2-8のようになります。いかがですか? これでモノクロ モードにおいてのアトリビュートエリアの操作が理解できたと思います。 いろ いろ工夫して、プログラムを作るときに応用してみてください。

表2-2 属性コード

Eyr	7	6	5	4	3	2	1	0	· 表示する
展性の 表示 する国副	0	0	アンダ ライン	アッパ	0	リバース	ブリンク	シークレット	属性コード
ブリンク	0	0	0	0	0	0	- 1	0	02H
リバース	0	0	0	0	0	1	0	0	04H
シークレット	0	0	0	0	0	0	0	1	01H
アッパライン	0	0	0	1	0	0	0	0	10H
アンダライン	0	0	1	0	0	0	0	0	20H
ブリンク・ リバース	0	. 0	0	0	0	1	1	0	06H
アッパ・アンダ ライン	0	0	1	1	0	0	0	0	30H
アッパ・アンダ	0	0	1	1	0	0	1	0	32H

	1		Text Mode ************************************
9988	START: EG	4-4 I MIX I U 0B900H	START ADDRESS
E826	MON: EC		:MONITOR ADDRESS
F3C8	CRIAD: ES		V-RAM START ADDRESS
F418	SKETA: EG		:V-RAM AV ADDRESS
0002	AVI: EG		BLINK DATA 62H
8884	AV2: EQ	U 94H	REVERSE DATA 84H
1000	AV3: EQ		SECRET DATA DIH
8618	AV4: EC		UPPER LINE DATA 18H
9626	AV5: EQ		: UNDER LINE DATA 20H
8886	AV6: EG	U Ø6H	: REVERSE AND BLINK DATA 06H
8030	AV7: EQ	C BEGG	: UPPER # UNDER LINE DATA 30H
8832	AV8: EQ		: UPPER & UNDER LINE & BLINK DAT
8888	CHG1: EQ		: 0(DEC:MAL DATA)
000A	CHG2: EG		:18(DECIMAL DATA)
3614	CHG3: EQ		(28(DECIMAL DATA)
081E	CHG4: EQ		(30 (DECIMAL DATA)
8928	CHG5: EQ		:48(DECIMAL DATA)
0032	CHG6: EQ		;50(DECIMAL DATA)
003C	CHG7: EG		:60(DECIMAL DATA)
8846	CHG8: EQ		;70(DECIMAL DATA)
0041	CHAR: EG)U 41H	:CHAR DATA 'A'
	OF	RG START	
B988 21C8F3	LE		
B983 1658	LE		: LINE DATA 88KETA
B905 3E41	LE	A.CHAR	:DISPLAY CHAR CODE

§5 モノクロモードでアトリビュートエリアを操作する

				95	モノグロモート ピアトリヒュートエリアを探り
	8987	77 23	Lonet:	L.D.	(HL).A
	8988	23		INC	HL
	8989	15		DEC	D
		C207B9		JP	D NZ.LOOPI
		2118F4			HL, SKETA
		3E00		LD	A.CHG1
	8912			LD	(HL),A
	B913	22		INC	LI CALL
	8914	250		1 10	HL A.AVI
	B916			LD	(HL),A
	B917	77			
				LD	A. CHG2
		3E0A		LD	K, ChG2
ı	B91A	77			
1	B91B	23		INC	HL
	891C	3E04		LD	HL A.AV2 (BD).A HL
	B91E	7.7		LU	(BL).A
	B91F			TINC	ML.
	B920	3E14		LD	A. CHG3
	B922	77		LD	A.CHG3 (HL).A HL
	B923	23		INC	HL
	B924	3E81		LD	A.AV3 (HL).A
	B926			LD	CHL1 - A
	B927	23		INC	HL
	8938	JE1E		LD	A.CHG4
	B92A			LD INC	(HL).A
	892B	23		INC	HL.
		3E10		LD	A.AV4
	892E	77		LD	(HL).A
	B92F	23		INC	HL.
	8938	3E28		LD	A . CHG5
	8932	77		LD	(HL),A HL A,AV5 (HL),A
	B933	23		INC	HL.
	8934	3E28		LD	A.AVS
	B936	77		L.D	(HL).A
	B937	23		INC	HL A.CHG6 CHLI.A
		3E32		LD	A.CHG6
	893A			L.D	CHL1.A
	B93B	2.3		INC	HI.
		3E06		0.1	A.AV6
	B93E	77		L.D	(HL),A
	B93F	23			
	8946	3E3C		1.D	A.CHG7 (HL).A HL
	B942			L.D.	(HL1.A
	B943			INC	HI
		3E38		LD	A AU7
	B946			LD	A.AU7 (RL).A
	B947			INC	H
		3E46		LD	A.CHG8
	B94A			LD	(HL).A
	B948			INC	
		3E32		LD	A.AV8
		77		LD	(HL1.A
	0348			20	THE TE
	ROAE	C326E8		JP.	MON ; MONITOR
	0391	02000		01	AMOUNT LOK
	B952			END	
	0.77				

カラーモードで アトリビュートエリアを操作

テキスト画面をカラー化する

PC-8801mkIISR は、スイッチを入れて BASIC が起動したときには、テキス | 画価のモノクロモードに設定されています。 BASIC でカラー画面にすること は無理なのですが、マシン語ではそう簡単ではありません。というのは、PC-8801mkIISR は、「pPD3301」という CRT コントローラを使用しているため に、DMA (Direct Memory Access) 操作をする必要があり、マシン語による テキスト画面のカラー化は非常に大変です。そこで、今回は BASIC のコマンド をマシン語で実行させる方法を使用して入ることにしました。

■ BASIC のコマンドをマシン■で実行させ■

原理は簡単です。BASIC の

SCREEN, CONSOLE, DEF, USR

をマシン語で実行させれば良いのです。

すると、BASICで打ち込まれ 図2-10 BASIC命令はテキストウィンドウへ

た BASIC 命令はそのままメモリ に記憶されずに、一旦中間「語に 変換されてから、テキストウィン ドウに入ります(mg-10)、この命 令を ROM 内のサブルーチンで呼 び出し、実行します。

具体的には、次の命令をBASIC で打ち込みます。



■リスト2-7 打ち込むBASIC命令■

10 SCREEN 1:CONSOLE 0,25,0,1:DEF USR=8HB900:A=USR(0)

66 カラーモードでアトリビュートエリアを操作

これからわかるように、カラーモードにしてからマシン語のプログラムのB 900日番地へジャンプすることになっています。しかし、マシン語のプログラム ではこのまま人力するのではなく、次のデータを8000日番地から入力します。

■リスト2-8 入力するデータ

これは、先程の BASIC プログラムが中間言語に直された結果のデータです。 800EH 豪地が12H でカラー両面に、11H でモノクロ両面になります。また、マンン語の実行器地は801AH と8019H 帯地にアドレスを入れることによって、どこからでもスタートすることが可能です。もちろん、このままでは実行できませんので、次のサブルーチンを呼び出します。

JP 087CH

これを利用したプログラムをリスト2-9に示します。 B920日 最地からの中間 活議を8000日 番地に結送してから "IP 0B7CH を実行するようになっていま す。 B932日 香地を11日にするとモノクロモードになります。また、ジャンプす る築地は B939日 に 下位バイト、B93A日に上位バイトを設定すれば、どこにで もジャンプすることができます。

TEVT COLOR SERVED

リスト2-9 テキスト カラー化

			1541	COLOR	-
		TEXT	/ カラー	カ ノ ルーチン	
8988 E626		START: MON:	EQU	⊕E626H	
B920				0B920H	
8000		DEST:			
ØB7C		RUN:	EQU	ØB7CH	
			ORG	START	
B963	212889 118886 812588		LD LD		
B989 B98A B98B	12	LOOP:	LD LD INC	(DE),A	
B98C 898D	13		1NC DEC	DE E	i
	C209B9 C37C0B		JP JP	NZ.LOOP RUN	
		1			

B914		DS	0CH ←	データの内容はともかく GCHバイト分娩保する
B928 88		DB	088	
B921 23		DB	23H	
B922 III		DB	00H	
B923 8A		DB	BAH	
B924 00		DB	ØBH	
B925 D3		DB	врзн	
B926 28		DB	20H	
B927 12		DB	12H	
B928 3A		DB	3AH	
8929		538	9DH	
B92A 20		DB	2BH	
B92B 11		DB	11H	
B92C 2C		DB	2CH	
B92D 8F		DB DB	0FH 19H	
B92E 19 B92F 2C			2CH	
B92F 2C B930 11		DB	11H	
B931 2C		DB	2CH	
B931 2C B932 12		DB	12H	: COLOR 12H,MONOCHROME 11H
B933 3A		DB	3AH	COLOR 1211, MONOCHRONE 1111
B934 97		DB	97H	
B935 20		DB	20H	
B936 E0		DB	BEBR	
B937 F1		DB	0F1H	
B938 6C		DB	9CH	
B939 50		DB	50H	: PROGRAM LOW ADDRESS
893A 89		DB	9B9H	: PROGRAM HIGH ADDRESS
B93B 3A		DB	3AH	
B93C 41		DB	41H	
B93D F1		100	ØF1H	
B93E E0		DB	REGH	
B93F 28		100	28H	
B948 11		DB	LIH	
B941 29		DB	29H	
B942 88		DB	86H	
B943 00		DB	BBH	
B944 IIII		DB	8.9 H	
0045		DS	BBH	
B945	;	US	ROH	
B950 C326E6	MAIN:	JP	MON	: MAIN PROGRAM
B953		END		

■ F419H 番地にデータを書き込んでテキストを8色に

Gコマンドを使用して B900H から実行してみてください。モニタのプロン プトが出てきました。このままでは本当にカラーモードになったのかわかりませんので、テストしてみましょう。

次のコマンドを実行させてください。

h] EF3C8 🗇

E コマンドで F3C8H 番地からの VRAM エリアのデータを表示させるものです。リスト2~10のように CRT 画面に VRAM のデータが表示されます。

A 11 7 62 - 10	*CC1C1*の実行禁風

F3CB				66	FF			98									F3CM FF
F3D9	38	38	138	83	46	28	30	39	20	46	46	29	36	30	29	46	88 FF 86 FF 88 F
F3E8	46	28	200	30	28	34	36	28	33	33	28	34	33	20	33	38	F 88 46 33 43 36
F3FB	20	32	300	28	34	36	28	34		28						201	28 46 46 28
F488	2111	29	28	FF	88	FF	99	FF	88	FF	88	46		43			FSCD
F418	46	46	28	28	300	28				E8				E8			FF
F428	100	E6	86	E8	Dist.	E8		E8		E8					BB		_0_0_0_0_0_0_0_0
F438		EB	80	E8	88	E8		E8		E8							_4_4_9_9_9_9_9_9
F448	46	33	44	300	100	33	3110				5.0		38				F3D0 30 36 20 46
F458	28	34	36	200	32	36			338	23	33		28				46 20 30 30 26
F468	34	36	28	34	IR	100	32		28	33			33		200		46 46 20 30 30 2
F478	38	28	34	36	26	5.3	28	26	26				38		46	46	8 46 88 FF
F488		36	38	28	46	836		36		28						20	88 FF 86 F
F498	88	E8	88	EB	88	E8	86	É6			86				88	EB	
F4A6	100	EB	100	EB	100	1837		83				E8		E8			_0_0_0_0_0_0_0_0
F488	88	Eβ	8.6	100	16.0	E8	88	ĒВ	46	33	45	36	28	34	36	28	

1行目のアトリピュートエリアの番地は 表2-3 カラーの属性コード F418H 番地から始まります。そこで、F419H 番地の属性コードにカラーデータを書き込み ます。それで1行目はカラー文字になるはず です。カラーの属性データは■2-3のようにな ります

F419H 条地にこの属性コードを書き込め は、任意の色を出すことができます。 もちろ ん。モノクロモードでやったように、1行の 文字を何色かに色分けすることもできます。



サブルーチンを使ったテキストのカラー化

今度は、カラー化で一般的であるサブルーチンを使用して。カラーモードに してみましょう。しかし、この方法では、バックグランドのコントロールはで きません。この方法は、前の方法と比べるとプログラム自体は短くなりますが、 バックグランドをコントロールできないのが残念です。

画面コントロール、エントリ・ポイント 6F6BH

これをただ CALL するだけではだめで、数個のバラメータが必要です。エン トリ・ボイントを CALL する前に、パラメータをセットしなければなりません。 バラメータは■-4(P.96)のようになっていますので、じっくり見てください、 また、プログラムはリスト2-11(P.96)となります。

表2-4 画面コントロール (6F6BH)の

E689H ··········テキスト・カラーマーク(FFH でカラー、00Hでモンクロ)

E6B4H にカラーコードをセットすれば任意の色を出すことができます。また、Eコマンドで F3C8H から表示させて、F419H 番地にカラーデータを入力すると、1行目の文字に色がつきます。カラーコードは(P.95)と同じです。

●リスト2-11 テキスト画面のカラー化(その2)

		*****	COLO	R TEXT MODE	****
		TEXT	カラーカ	ノ ルーテン (CAL	L 6F6BH)
B900 E626 6F6B		START: MON: COLOR:	EQU	8B989H 8E626H 6F6BH	:モニター ノ エントリー M* トイント :カラー モート* ルーチン
			ORG	START	
B902 B904 B906 B909 B90B	8658 8E19 3E81 32B2E6 3E19 32B3E6		LD LD LD	B.58H C.19H A.01H (@E6B2H).A A.19H (@E6B3H).A	:CRT / ケタスウ ヺ セット :CRT / キーロスク ヺ セット :スクロール / カイシ / キーョウ ヺ セット :スクロール / オワリ / キーョウ ヺ セット
B90E B910	3284E6		LD	A.0E8H (0E6B4H).A	:カラーコート" ヲ セット
	3E88 3288E6		LD	A.88H (@E6B8H),A	:ファンクション キー / ヒョウシ" / ON OF
B918 B91A	3EFF 3289E6		LD LD	A,8FFH (0E689H),A	: カラーマーク ノ セット カラーマーク FFH
B91D	CD6B6F		CALL	COLOR	
B928	C326E6	;	JP	MON	
B923			END		

バッググランド・カラーを変える

キャラクタ単位のカラー化はできましたので、今度はバックグランドのカラー化について考えてみましょう。

 $_{N \to D}$ グランドのカラーコントロールは、 $_{N \to D}$ が行うようになっています。 $_{N \to D}$ パートの 52H にカラーデータを入力すればいいのです。カラ

ーデータは■2-5のようになっています。

このテストはモニクラマンドを使用することによって簡単にできます。"O"コマンドを 使用すればよいのです。ですから、たとえば 糸を扱示したいのであれば。

h] 052, 20 日 赤のデータ

とすればバックグランドの色は赤に変ります。いろいろなデータで嫌めてください。な お、バックグランドの色が行になった場合は、な お、バックグランドの色が行になった場合は、 技術されているデータは読むことができません。これは、キャラクタの色とバックグラン ドの色が個になるなかがで

ブロック転送命令を使う

リスト2-8のプログラムではB920日番地からのデータを8000日番地に転送しています。この転送のために使用される命令は、全部で9命令ですが、4命令にすることができます。

Z-80には、プロック転送命令という強力な命令があります、このブロック転送命令は6命令を1命令にしたようなもので、なおかつ6命令を使用するより速く処理できるものです。この命令は次のようになります。

LDIR

書式 LDIR

機能 HL ペアレジスタで指定される番地のデータを DE ペアレジスタ で示される番地に。BC ペアレジスタが 0 になるまで転送します。

(転送元のアドレスをセットする) (9) ID HL. OB920H LD DE, 8000H (転送先のアドレスをセットする) LO BC. 0025H (転送するバイト数をセットする) LDIR

■ LDIR を使った場合と使わない場合を比較すると

LDIR 命令は、実行する前に各ペアレジスタにデータをセットしてからでな いと実行することはできませんが強力な命令です。ためしに、この命令を使用 した場合と使用していない場合を比べてみましょう。

共通

LDIR を任	使わなかった場合	LDIR &	使っ	た場合	
LD	HL, OB920H	LD	HL,	OB920H	
LD	DE, 8000H	LD	DE,	8000H	
LD	BC, 0025H	L,D	BC,	0025H	
LOOP : LD	A, (HL)	LDI	R		
LD	(DE), A				
INC	HL				

と、このようになります。これだったら、使用しないと損ですね。どんどん使 っていくことにしましょう.

この LDIR 命令を使用したプログラムをリスト2-12に示します。リスト2-9 と動作内容は全て同じです。

■リスト2-12 LDIRを使ったテキストカラー化 ; **** TEXT COLOR ***** TEXT / カラーカ / ルーチン START: FOIL ивравн RES26H MON: EQU EQU SRCE:

INC DE DEC B JP NZ. LOOP

> EQU RUN: ØB7CH ORG START

8020H

§6 カラーモードでアトリビュートエリアを操作

B983 B986 B989	212989 118888 812588 EDB8 C37C8B		LD LD LD LDIR	HL.SRCE DE,DEST BC.0825H RUN	
B98E	COVCBD		DS	12H	
8925 8926 8927 8928 8929 8920 8920 8920 8921 8930 8931 8933 8933 8933 8933 8933 8933 8933	23 88 80 80 80 80 80 80 80 80 80 80 80 80		DB D	82H 823H 828H 828H 828H 828H 828H 828H 8	: COLOR 12H.MONOCHROME 11H :PROGRAM LOW ADDRESS :PROGRAM HIGH ADDRESS
B950	C326E6	MAIN:	JP	MGN	: MAIN PROGRAM
B953			END		

TEXT文字とバックグランドの カラー化デモプログラム

デモプログラムの仕様

ノモノロノノムの江南

TEXT 文字とバックグランドカラーのカラー化が理解できたと思いますので、仕上げの意味も含めて、カラーのデモプログラムを作ってみましょう。

図2-11のような画面のレイアウトを考えていきましょう。まず、プログラム の仕様をはっきり決めておかなければなりません。プログラムを作る前に仕様 を決めておかないと、初めの目的がわからなくなってしまうからです。

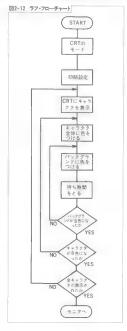
プログラムの仕様

- a。同一キャラクタを画面全部に表示させる(80×25モード)
 - b, キャラクタを"→" (1CH) ~"秒" (F7H) まで変化させる
 - c、キャラクタの色を順次にB色表示させる
 - d. キャラクタの色 | 色に対して、バックグランドの色を順次 8 色にする
 - e. 全部表示し終ったら、モニタに戻る

大まかなフローチャートを考えると図2-12のようになります.

図2-11 画面のレイアウト A ++9クタを --*(1(d)) から **が*(77ii) まで 変化させる 文字の色を構次 またに変えていく 原文を色にする

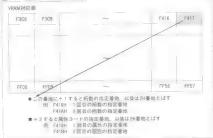
■7 TEXT文字とバックグランドのカラー化デモプログラム



全体のキャラクタに色をつけるには

CRT 全体に色をつけるためには、アトリビュートエリアの感性コードに対応 する基地に、感性コードを書き込む必要があります、行縁の VRAM 対応表を見 てください、これには、アトリビュートエリアの対応表はありませんが、感性 コードに対応する影響はすぐわかります。 園―13のようになります。

図2-13 属性コードに対応する書地



したがって、1桁ごとの属性コード指定番地へ属性コードを書き込まなければなりませんので、全部で25回の作業が必要となります。しかし、下の行までの番地は120パイト分あるので、アドレスに78H(120)分を加えれば済みます。

■カラーの属性コードは 20H とび

カラーの属性コードは08日~E8日まで変化するのですが、都介悪いことに達 続になっておらず、20日とびになっています。したがって、初めのデータに20 日加えて属性コードを変えてやらなければなりません。

バックグランドに色をつけるには

プログラムでバックグランドに色をつける方法は、I/O ボートの52H ヘカラ ーデータを送ればよいのです。カラーコードは00H~70H まで変化しますが、 これも10H とびになっていますので、初めのデータ00H に10H ずつ起していけ ばよいことになります。

例 初めのデータ DDH

2回目のデータ 10H (00H+10H) 3回目のデータ 20H (10H+10H)

8回目のデータ 70H (60H+10H)

■旧8801ならばボーダカラーも操作できる

ただし、旧 PC-8801をお持ちの方は、ボーダカラーも操作することができます。 下位3 ピットがボーダカラーのピットですので、この3 ピットに0~7 H までのデータを書き込みます。

例

11H ボーダカラーのデータ (青色になる) バックカラーのデータ (青色)

この例では、CRT の画面全体が青色になります。なお、mkII や SR では残念ながら使用できません。

プログラムはこうなっている

では、リスト2-13にテキスト画面のカラー化デモプログラムを示します。これから、プログラムリストの説明をしていきましょう。

●リスト2-13 テキスト■■ カラー化デモプログラム

: **** TEXT COLDR ***** :PROGRAM NAME --> ctdemo.e :TEXT / カラーノ デモプログラム START: EQU @B980H EQU BE626H SPCF: EDIT 8B928H FOL RARRH RIIN: EOU RR7CH ив7С ADCRT: EQU 0F3C8H

	ORG	START	
	CRT MODE F	ROGRAM	
B908 2120B9 B903 110000 B906 012500 B909 EDB0 B908 C37C0B	I.D	HL.SRCE DE.DEST BC.0025H RUN	
BORE	DS	12H	
8824 #8 8924 #	DB D	00H 23H 00H 00H 00H 00H 00H 00H 00H 00H 00H 0	: COLOR 12H.MONOCHROME 11H :MAIN PROGRAM LOW ADDRESS :MAIN PROGRAM HIGH ADDRESS
B948 11 B941 29 B942 88	DB DB	11H 29H 00H	
8943 IIII 8944 88	DB DB	H08	
B945	DS	0BH	
	MAIN PROG		
B953 3EIC B955 324DB9 B958 3E08			:CURSOR OFF :CHR START DATA :CHR DATA AREA

§7 TEXT文字とバックグランドのカラー化デモプログラム

```
B95A CDA7B9 SMAIN: CALL DISP
B95D 8608 LD B.08H
B95F 2119F4 CCOLR: LD HL.0F-
                                           HL, 0F419H
B962 117800 LD DE.0078H
B965 @E19
                                    LD C.19H
B967 77
                      CLOP1: LD
                                            (HL).A
B968 19 ADD HL.DE
B988 1-
B969 8D
B968 C267B9 :
BCOLR: PUSH BC
PUSH AF,
                                    JP NZ, CLOP1

        B96D C5
        BCOLR: PUSH BC

        B96E F5
        PUSH AF

        B96F 3600
        LD A.80H

        B971 0610
        LD B.10H

        B973 0808
        LD C.08H

        B975 D352
        BLOP1: OUT (52H).A

B977 CDCCB9 CALL WAIT
B97A 88
B97B 0D
                                    ADD A.B
                            DEC C
JP NZ.BLOP1
POP AF
POP BC
PUSH BC
LD B.28H
ADD A.B
POP BC
BC B
JP NZ.CCOLR
PUSH A (8894DK)
CP BF8H
JP NZ.SMJ
                                  DEC C
897C C275B9
897F F1
8988 C1
B981 C5
 B982 8628
B984 88
R985 C1
B986 Ø5
B987 C25FB9
 R98A F5
B98B 3A4DB9
B98E FEF8
B998 C2A3B9
B993 CD9842
                                  CALL 4290H
                                                                            CURSOR ON
 B996 3E00
                                 LD A.80H
OUT (52H).A
                                                                            :BACK COLOR TO BLACK
 B998 D352
 B99A 3E8C
                                LD A. 0CH
CALL 0818H
POP AF
                                                                   :CRT CLEAR DATA 0CH
:CRT CLEAR SUB
 B99C CD1800
 BOOF FI
                                    JP.
 B9A@ C326E6
                                             MON
                        SMJ: POP AF
 89A3 F1
 B9A4 C35AB9
                                    JP SMAIN
                        ;DISP SUB -----
 B9A7 E5
                        DISP: PUSH HI
 B9A8 D5
                                    PUSH DE
 B9A9 C5
                                    PUSH BC
 BSAA F5
                                 PUSH AF
LD HL

        B9A8 21C8F3
        LD
        HL.ADCRT

        B9A6 21C8F3
        LD
        DE.0828H

        B9A6 112800
        LD
        C.19H

        B9B1 0E19
        LD
        C.19H

        B9B3 0858
        DLOP2: LD
        B.58H

        B9B5 3A4D89
        LD
        A.70894DH)
```

```
B9B8 77
              DLOP1: LD (HL).A
B9B9 23
                       INC
                            HL.
B9BA 85
                       DEC
B9BB C2B8B9
                            NZ. DLOPI
89BE 19
89BF 8D
                       ADD
                            HL, DE
                       DEC
B9C0 C2B3B9
                       JP
                            NZ.DLOP2
B9C4 324DB9
                            (#894DH),A
                            AF
                            BC
                            HL
                       RET
               :WALT SUB -----
Bacc cs
                       PUSH BO
                       PUSH AF
BOCE ØEDI
                            C. BIH
BODO BEAR
               WL3:
B9D2 3EFF
               W1.2:
                            A. ØFFH
                       DEC
               WL1:
                            Α
                       JP.
                            NZ.WLI
B9D8 05
                       DEC
B9D9 C2D2B9
                            NZ, WL2
                       DEC
B9DD C2D8B9
                       JP
                            NZ. WL3
B9E0 F1
                       POP
                            AF
B9E1 C1
                            BC
B9E2 C9
```

■ここで使用している ROM 内ルーチン

CRT のモード設定が終った後の B950H 番地から、メインプログラムに入っていますが、少し簡単にするために ROM 内ルーチンを使用しています。

ROM 内ルーチンは次のようになります。

RUMの香地	サフルーチンの意味
428BH	→カーソル表示を消す
4290H ·	→カーソルを表示する
0018H	
	入れる必要がある

■メインプログラム内での汎用レジスタの■途

B94DH 番地は、キャラクタコードをストアしておくワークエリアになります。これはレシスタがすべて使用されているために必要とするものです。

次に、メインプログラム内での汎用レジスタの用途について述べることにします。

§7 TEXT文字とパックグランドのカラー化デモプログラム

HI ベアレジスタ アトリビュートエリアの属性コード番地

DE ペアレジスタ 真下の行に持ってくるためのデータ(120パイト分)

Bレジスタ キャラクタの色の個数 Cレジスタ 行数を数える

A レジスタ カラーコードデータ

これ以外の用途で使用される場合は、すべて PUSH 命令で逃避させています。

最後のキャラクタコードのF7Hの表示が終ったならば、カーソルを表示して、バックグランドの色を無にしています。というのは、最後のバックグランドの色が打になると、プロンプトがどこにあるかわからなくなるからです。最後に価値をクリアして、モニタに戻ります。

このプログラムは、終了するまでに約1時間かかります。"WAIT"のサブルーチンのデータを変えれば、もう少し遠くなりますが、遠くなりすぎてバック クランドの色を変化させるときにチラツキが放れてしまいます。6し、遠くし たい人は、BBDIH 番地のデータ AOH を変えて小さい流にしてください。

■参考―テキスト画面のカラー化デモプログラム その■

このプログラムは、「胸部コントロール・サブルーチン」(リスト2-13)を使用したもので、リスト2-13と同じように伴ってあります。ただし、バックグランドのカラー化はできないので含いてあります。また、WAIT時間(接示している時間)は接くしないとはっきり見えませんので、少し、接くしてあります。

●リスト2-14 テキスト画面のカラー化デモプログラム その 2

:**** TEXT COLOR ***** : :PROGRAM NAME --> csdemo.e

: :TEXT / カラー/ デ`モプログ`ラレ ソノ 2 sub(CALL 6F6BH)

B988 START: EQU 08980H E626 MON: EQU 8E626H F3C8 ADCRT: EQU 0F3C8H

B988 8658

B982 8E19

B984 3E81

ADCRT: EQU 0F3C8H COLOR: EQU 6F6BH :CRT COLOR SUBROUTINE

ORG START

:CRT MODE PROGRAM -----

LD B.50H (ヨコ / ケタスウ ヲ セット トル 0(8レジ スタ)80 LD C.19H (タテ / キョウ スウ ヲ セット スル 0(Cレジ スタ)25

LD A,01H :スクロール カイシ キーヨウ

■2章 モノクロ画面からカラー画面への招待

```
8965 3282E5 LD (6E682H).A

8989 32E19 LD A.1991 (220-1-シェクリョウ・ギョウ

8986 2282E6 LD (6E683H).A

8918 3284E6 LD (6E683H).A

8918 3284E6 LD (6E685H).A

8913 3284E6 LD (6E685H).A

8915 3288E6 LD (6E685H).A

8916 3288E6 LD (6E685H).A

8916 3288E6 LD (6E685H).A

8916 3288E6 LD (6E685H).A

8916 3288E6 LD (6E685H).A
  B928
                                   DS #1H ; CHR DATA AREA
                          : MAIN PROGRAM-----

        B921 CD8842
        Main:
        CALL 428BH
        :CURSOR OFF

        B924 3EIC
        LD
        A.ICH
        :CHR START DATA

        B926 3228B9
        LD
        (B929H).A
        :CHR DATA AREA

                                 LD A.08H
  B929 3E88
B92B CD63B9 SMAIN: CALL DISP
 B942 8628
B944 88
B945 C1
B946 85
                                 LD B.28H
ADD A.B
POP BC
 8946 85 POP BC

8947 C23089 P NZ.CCOLR

8948 3.2289 POSH 4 (68928H)

8948 3.2289 POSH 4 (68928H)

8948 5.2289 P NZ.CCOLR
 B953 CD9642
                                 CALL 4290H
                                                                         CURSOR ON
 B956 3E0C
                                LD A.ØCH
CALL ØBI8H
                                                                     : CRT CLEAR DATA 8CH
: CRT CLEAR SUB
 B958 CD1808
 B95B F1
                                   POP AF
 B95C C326E6
                                   JP MON
 B95F F1 SMJ: POP AF
 B960 C32BB9
                          JP SMAIN
                         ; DISP SUB -----
8963 E5 DISP: PUSH HL
8964 D5 PUSH DE
8965 C5 PUSH BC
8966 F5 PUSH AF
8967 21C8F3 LD HL.ADCRT
8964 112888 LD DE.8028H
```

§7 TEXT文字とバックグランドのカラー化デモプログラム

```
896D ØE19
                            C. L9H
B96F 0650
               DLOP2: LD
                            B. 50H
B971 3A28B9
                            A. (0B920H)
               DLOP1:
                            (HL), A
                            HI
                       DEC
B977 C274B9
                       JP
                            NZ. DLOP1
                       ADD
                            HL.DE
B978 80
                       DEC
B97C C26FB9
                       JP
                       INC
B988 3220B9
                            (88928H).A
B983 F1
                            AF
                       POP
8984 CI
                            BC
B985 D1
                            DE
8986 E1
                            HL.
                       RET
              :WAIT SUB ----
B988 C5
               WAIT:
                       PUSH BC
B989 F5
                       PUSH AF
B98A BEB2
                            C.82H
898C #6FF
              WL3:
                            B. ØFFH
ROBE SEEF
               WL2:
B998 3D
               WLI:
                       DEC
B991 C290B9
                            NZ,WLI
                       JP
                            NZ.WL2
                       JP.
B999 C28CB9
                            NZ.WL3
B99C F1
                            AF
8990 C1
                            BC
B99E C9
899F
                       END
```

2 準ではテキスト画面とバックグランドをカラー化する方法を中心に該を進 かでもましたが、いかがでしたか? テキスト画面をカラー化する方法はいろい ろあるのですが、これが一番簡単だと思います。テキスト画面をカラー化して いろいろと乗しんでください。





キャラクタ単位のカラー化ができまし たので、この章ではキーボードからデー タを読み取る方法を考えてみましょう。

PC-8801mkIISRでは、8ビットデータ の各1ビットごとに1個のキーが割り出 てられているので、キーが押されたかど

向. 8方向に移動させるプログラムを作 また、ちょっと耳慣れない「闘理演算

命令。についても勉強しましょう。この 命令を使うとドット単位で両面を動かす こともできるので、キャラクタをスム

ズに移動でき、大変便利です。

キーボードetc.に使われる I/O命令

I/Oポートから8ビットデータをとり込むIN命令

I/O とはよく使用される言葉です。I/O は Input / Output の略であること は、翻発知のことと思います。I/O は本来、メモリ以外へ入出力することが目的 となっていました。しかし、今はI/O を利用してメモリバンクなどによる増放 等。1 からいろと利用されています。したがって、パソコンをうまく利用するた かには、I/O 命令を利用してI/O ボートをアクセスする必要があります。 Z-80 では、I/O ボートは今部で FFH (256) まで利用できます。まず、IN 命令につ いて説明していきましょう。

IN 命令は、I/O ポートから 8 ピットデータを取り込む命令です。キーボード からデータを取り込んだり、フロッピィディスクからデータを取り込んだりす るのに多く使用されます。

IN A. (N)

書 式 IN A. (8ビットのポート番号)

■ 8ビットポート番号で指定されたボートから、Aレジスタにデータを取り込みます。

IN A. (00H)

ポート番号00HからAレジスタにデータを取り込みます。

■ IN 命令 (C レジスタ ■ ■)

もういとつ IN 命令があります。それはCレシスタ側接と言われるもので、C レジスタのデータを I/O ボート番号と見なして 実行するものです。C レジスタ は演算できますので、ボート番号が連続しているときなどは、非常に便利です。 また、取り込むレジスタは、A レジスタばかりでなくすべての批用レジスタが

第1 第 キーボードを操作する

利用できます。

IN r. (C)

書 式 IN r. (C) (rはA, B, C, D, E, H, Lレジスタ)

■ 能 Cレジスタで指定される I/O ポートから, r で指定されるレジス タにデータを取り込みます

例 LD C, 01H

IN D. (C)

ポート署号の01Hから、Dレジスタにデータを取り込みます。

フラグ フラグはデータの値により、次の3つが変化します。 P/V フラグ:入力データの入っているビットが偶数のとき1。

奇数のとき 0

S フラグ: 入力データの最上位ビット $(7ビット \ B)$ が1のとき 1, 0のとき 0

Ζフラグ:入力データが 0 のとき 1、 それ以外のとき 0

この命令は、どのキーボードが押されたかを調べるときに使用されています。

I/Oポートにデータを出力するOUT命令

OUT 命令は IN 命令とは逆に、I/O ボートにアータを出力する命令です。し かし、この命令はデータを無視して、I/O ボートのスイッチとして使用されるこ ともあります、PC-8801mkIISRでは、カラーバレットや RGB のカラーブレー ンなどに利用されています。

OUT (n), A

■ 式 OUT (ポート番号), A

■ 能 ポート番号で指定されたポートに、Aレジスタのデータを出力 します。

例 OUT (52H), A

ポート番号のII2HへAレジスタのデータを出力します。

OUT 命令にも、IN 命令と同様にCレジスクを使用した間接命令があります。

■ OUT 命令 (C レジスタ■■)

OUT (C), r

■ 式 OUT (C), r (rはA, B, C, D, E, H, Lレジスタ)

能 ボート番号で指定されたボートに、rで指定されるレジスタの データを出力します

(9) LD C, 52H

OUT (C). III

ポート番号の52HへBレジスタのデータを出力します。

なお、この命令では、フラグは変化しません。

I/O 命令はキーボードからの取り込みなどに多く使用されているので、応用のし方をしっかり覚えましょう。

キーボードから データを読みとるには

ビットを調べればキー入力がわかる

こう[を励いればな十一人/カカイルの

PC-8801mkIISRはCPU(Z-80A)でキーボードを1 側 : 側調べているので、 キーボードを調べるプログラムをマシン語で作るのは比較的家にできます。キ ーボードのキーの数は全部で90側ありますが、キーを8個すつのデータに分け て、8ビットのデータとしてI/O 命令で読み取ります。全部のキーを調べるた めには、I/O 命令を12回 (0BH) 使用しなければなりませんが、実際のプログ ラムでは、調べるキーの数は多くで 610-20くらいでしょうから、I/O 命令を多 く使用することはないでしょう。SR では、8ビットデータの各1ビットごと に、『個のキーが削り当てられています(図3-1、P117).

■各ビットの状態とキーの状態

各ビットを調べれば、8個のキーの内どのキーが押されたかがわかります。 キーの状態によって各ビットは次のように変化します。

各ピットの状態 キーの状態 1 押されていない

図3-2 キーボード・データとビットの対応関係



押されている

押されているときが1の状態で、押されていないときが1の状態になっています。少し見にくいのですが、ソフトで何とでも変更できるので、気にする必要はありません。

図3-1 キーボード・データ

۲» ا/۵ ۲									- 83				5				4				3			2						0	
P	ŭ.	- 2																													
0	0	Н							6				5				ñ				3			10				1		0	
0	1	Н			4												=				+			4		-1		9		8	
0	2	Н			0 1	0.00			F,				EY	4			D				0			80		1		AF		(2	
0	3	Н			0	9			200				M				Lij				ĸ			1		ĺ				H Ź	
0	4	Н			4.5	y			100				Ų,				Th				S			E Z				0.9		P to	
0	5	Н			- 1	- ,	51	-		`~				Z		¥	I					r		2				Y		×	
0	6	Н			7 '	4				r 7t 1		5	%	X		4	\$	ウゥ		3 :		7	2		7		1	1	F	7	7
0	7	Н				- 1	2	/	1	1			>	<i>J</i> L 0			<	*			+	L		16	4		9		33 8	3 (
	8	Н	F-		0 1	R		97	F	1	İ	ħ	t		:	G	RP	Н		I D	N E	S						Ť		HOM	
0	9	Н			E S	S		50	8	A		4		5		f		4		f		3	1		S		f		T	STO	2
0	A	Н		1	A	PC	S						-			С	01	PΥ	1	H 8	EL	. Р		4				l		ΓA	В
0	В	Н		- F	20	L	L	: R	000	LL																					

対応ビットが0なら、キーは押された

■1を見ながら具体的に説明していくことにしましょう。キーが押されていないときはどのデータのビットも、すべて1になります。つまり、8ビットデータはFFHになります。

もし、どこかのキーが押されていれば、いずれかのビットがⅡになっていま す。つまり、FFH 以外のデータになるわけです。

したがって、目的のキーが行されたかどうかを調べるには、目的のキーのある1/0 アドレスを指定して、そのキーに対応するビットが0 になっているかを 調べれば良いわけです、具体的に表すと、図3-3のようになります。

図3-3 1/0アドレス((0日のときの)単体的なデータ(9)

				Ľ,	ット				
/Oアドレス DOHのとき	7	6	5	4	3	2	1	0	
キーが押されていないとき	1	1	1	1	1	1	1	1	FFH
5 のキーが押されたとき	1	1	0	1	1	1	1	1	DFH
1 のキーが押されたとき	1	1	1	1	1	1	D	1	FDH
1と 1 のキーが押されたとき	1	1	1	1	0	1	D	ī	F5H

■ A のキーが押されたかを聞べる

たとえば、"A"のキーが押されたかどうかを調べるプログラムを作ってみま しょう、 III-1をもう一度見てください、"A"のキーは I/O アドレス [02H] の1ビット目にあります。したがって、このビットが0 であれば "A"のキー が叩されたことになります。プログラムは次のようになります

KEY: IN A, (02H) : A のキーのある I/O アドレス CP 0FDH : A のキーのデータ IIIIII 01B JP Z, MON : A のキーが押されたらモニタへ戻る CALL WAIT : 時間待ちのサブルーチンを呼ぶ: IP KEY

31 1121

WAIT : LD | OFFH

; 時間待ちのサブルーチン

LOP : DEC B

JP NZ, LOP

RET

このように、簡単なプログラムで済みます.

■ D のキーを調べ ■ の変更点

もし、"A" のキーから、"D" のキーに変更するのであれば、データを次の ように変えれば良いことになります。

Aのデータ

Dのデータ

11111101 B (FDH)

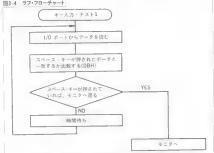
(FEH)

つまり、FDH を EFH にするだけです。

キー入力のテストプログラムを作る

キー入力のテストとして、スペースキーが押されたらモニタへ戻るプログラ ムを作ってみましょう。ラフ・フローチャートを図3-4に示します。

図3-4 ラフ・フローチャート



プログラムは、**リスト3-1**のようになります。新しい命令は使っていませんので、理解出来ると思います。

●リスト3-1 キー入力のテスト

			***	KEY IN T	EST ********
		: PROGR	AM NA	ME>k	eyt1.e
		: +- ===	ウリョク	J 721 7°	07*94 (CP @BFH (
		: ****	****	*****	新英斯英佩佛斯斯伯斯斯特拉尔哈特
E626 B988		MON: START!	EQU EQU	0E626H 0B900H	
			ORG	START	
B902	DB09 FEBF CA26E6	MAIN:	EN CP JP	A.(89H) 8BFH Z.MON	(1/0 本*-ト / 89H カラ デ*-タ リ ヨミトル (スペ*-ス キー カ**オサレタテ*-タト イッチ スルカヒカクスル (スペ*-ス キー カ** オサレデイレバ** モニター ヘ モト*ル
B987 B989 B96B	86FF 85	LOP2: LOP1:	LD LD DEC	B. ØFFH B	:シ`ガン マチ ノ ルーチン
B90F	C208B9 0D C209B9		JP DEC	NZ,LOP1 C NZ,LOP2	
8913	C30089		JP	MAIN	
B916			END		

論理演算命令を使って プログラムを作る

論理演算命令は、大きく3

リスト3-1で分岐条件を作っているのが、CP(コンペア)命令です。この命令 は非常に便利で、大きい、等しい、小さい、が区別出来るものでした。今度は、 CP 命令の代りに「論理演算命令」と呼ばれる命令を使用してみることにしまし 、

まず、論理演算命令について説明しましょう。

論理演算命令は、1ビットごとの論理演算を行うもので、マシン語では大切な 命令のひとつです。論理演算命令は、大きく分けて次の3命令に分けられます。

- 1. AND
- 2. OR 3. XOR

では、まず一番多く使用される AND 命令からみていきましょう。

かけ算みたいなAND命令

論理演算だからといって、難しく考える必要はありません。数学の乗算と同 じに考えて構いません。■3-1の計算をみてください。

表3-1 AND命令の計算例

А	×	В	22	Х
0	×	0	-	0
1	×	0	=	0
0	×	1	=	0
1	×	1	=	1

このようになります。お互いに1のときだけ答は1になります。0があれば

答は0となります。2進法ですから、0と1以外の数字はありません。

具体的な例で説明しましょう。36H と FAH との AND の論理演算をしてみましょう。

■3-2 36HとFAHのAND

ビット数	(36H)		(FAH)		演算結果
	A	×	8	=	Χ
7	0	×	1	-	0
6	0	×	1	=	
5	1	×	1	-	1
4	1	×	1	-	1
3	0	×	1	=	0
2	1	×	0	-	0
1	1	Х	1	=	1
0	0	×	0	=	0

演算結果が32Hとなりましたが、大切なのは各1ビットです。 なお AND 命令には、次の3種類があります。

■ 3 つの AND 命令

AND r

式 AND r (rはA, B, C, D, E, H, Lレジスタ)

■ 能 Aレジスタと r で指定されたレジスタとで、各ビットごとに AND 演算を行い、その結果を A レジスタにセットします。

AND B

AレジスタのデータをA3H、BIレジスタのデータを9BHとす

ると,

A レジスタのデータ…… 1 0 1 0 0 0 1 1 B B レジスタのデータ…… 1 0 0 1 1 0 1 1 8 演算結果…… 1 0 0 0 0 0 1 1 B

となります。

フラグ Sフラグ:命令を実行した結果,

の変化 最上位のビット (7ビット目) が1なら↑

最上位のビット (7ビット目) が □ なら Ⅱ

§3 ■門演算命令を使ってプログラムを作る

となります.

Zフラグ: 命令を実行した結果が

00H のとき1

サカ以外のと■ O

となります

P/V フラグ:命令を実行した結果、1になっているビットの数が

偶数なら1

奇数なら0

となります.

C フラグ:常に D となります。

AND n

■ 式 AND n (n=0~255)

■ 能 Aレジスタとデータ■とで各ビットごとに AND 演算を行い、その結果を Aレジスタにセットします。

AND 78H

A レジスタのデータが A3H. データ n を78H とすると.

Aレジスタのデータ……10100011B

となります。

フラグ Sフラグ: 含念を実行した結果

の変化

最上位のピット (7ビット目) が 1 なら1

最上位のピット (7ビット目)が日なら0

となります。

Z フラグ: 命令を実行した結果が 00H のとき 1

それ以外のとき目

となります.

P/V フラグ:命令を実行した結果, 1 になっているピットの響が

偶数なら1

奇数なら0

となります。 Cフラグ: 常にOとなります

AND (HL)

書 式 AND (HL)

機 能 A レジスタと(HL) で指定された番地のデータとで、各ビットご とに AND 消算を行い、その結果を A レジスタにセットします。

(AND (HL)

A レジスタのデータが A3H, HL ペアレジスタが指定する番地の データを 3AH とすると.

Aレジスタのデータ……10100011B

(HL) のデータ………001110108 論算結果………10100108

となります.

フラグ Sフラグ:命令を実行した結果, の変化 最上位のビット (7)

最上位のビット (7ビット目) が1なら1

最上位のビット (7ピット目) が0なら0

となります。

Z フラグ:命令を実行した結果が

00 H のとき 1

それ以外のとき0

となります.

P/V フラグ:命令を実行した結果。1 になっているビットの画が

偶数なら1 奇数なら0

となります。

Cフラグ:常に Dとなります。

たし算みたいなOR命令

この OR 命令は、数学の加算と同じに考えてかまいません。■3-3の計算式を

みてください。

■3-3 OR命令の計算例

^
U
1
1
1

このようになります。どちらかが1であれば、答は1になります。この場合 は、両方とも0のときだけ答が0になります。具体的な例で説明しましょう。 36HとFAHとのORの論理消算をしてみましょう。

■3-4 38HとFAHのOR

ピット数	(36H)		(FAH)	演	算結果
	A	+	В	-	Х
7	0	+	1	-	1
6	0	+	1	=	1
5	1	+	1	-	1
4	1	+	1	-	1
3	0	+	1	=	1
2	1	+	0	=	1
1	1	+	1	=	1
0	D.	+	0	=	0

演算結果が FEH となりました。この命令は、任意のビットを1にしたいとき に多く使用されます。

■ 3 つの OR 命令

なお、OR 命令には、次の3種類があります。

OR r

- 式 OR r (rはA, B, C, D, E, H, Lレジスタ)
- 能 Aレジスタと r で指定されたレジスタとで、各ビットごとに OR 演算を行い、その結果を Aレジスタにセットします。

(9) OR B

■ 3 章 キーボードを操作する

A レジスタのデータが A3H, Ⅲ レジスタのデータが 9BH とする

2

A レジスタのデータ……10100011B

■レジスタのデータ……100110118 演算結果……101110118

となります。桁上がりは無視されます。

フラグ Sフラグ:命令を実行した結果が

最上位のビット (7ビット目) が1なら1 最上位のビット (7ビット目) が0なら0

となります。

Z フラグ:命令を実行した結果が

00 Hのとき1

それ以外のとき 0 となります。

P/V フラグ:命令を実行した結果、1 になっているビットの響が

偶数なら1 条数なら0

となります

C フラグ:常に 0 となります.

OR n

の変化

■ ± OR n (n=0~255)

■ 能 Aレジスタとデータ n とで各ビットごとに OR 演算を行い、その結果を A レジスタにセットします。

■ OR 78H

Αレジスタのデータが A3H、データ n を78 H とすると、

Aレジスタのデータ……10100011B データn………01111000B

演算結果-----111110118

となります.

フラグ Sフラグ:命令を実行した結果が

の変化 最上位のビット (7ビット目) が1なら1

§3 論理演算命令を使ってプログラムを作る

最上位のピット (7ピット目) が0なら ||

となります.

Z フラグ: 命令を実行した結果が

00 H のとき 1

それ以外のとき 0

となります.

P/V フラグ:命令を実行した結果。 1 になっているビットの皿が

偶数なら1

奇数なら 0 となります。

Cフラグ:常に0となります.

OR (HL)

■ 式 OR (HL)

能 Aレジスタと(HL)で指定された番地のデータとで各ビットごとに OR 演算を行い。その結果を Aレジスタにセットします。

OR (HL)

A レジスタが A3H, HL ペアレジスタが指定する暑地のデータを 3AH とすると、

Aレジスタのデータ……10100011B

(HL) のデータ………00111010B 清算結果………10111011B

となります.

フラグ Sフラグ:命令を実行した結果,の変化

最上位のビット (7ビット目) が1なら1 最上位のビット (7ビット目) が1なら0

となります。

Z フラグ:命令を実行した結果が

00H のとき 1 それ以外のとき 0

となります.

P/V フラグ: 命令を実行した結果.1になっているビットの■が

偶数なら1 奇数なら0 となります。 Cフラグ: 常に0となります。

等しければ0, XOR命令

この XOR 命令は、数学でたとえるなら何でたとえれば良いでしょうか、ちょっと思い浮かびません。この命令は、対応するでつのビットが非に等しければ 0 となり、違っていれば1となるものです。■3-5の計算式を見てください。

表3-5 XOR命令の計算例

Α	?	В	-	Х	
0	?	0	=	0	I
0	?	1	=	1	ı
1	?	0	=	3	ı
1	?	1	=	0	ı
					1

このように、両方のビットが等しければ答は0になり。違っていれば答は1 になります、具体的な例で説明しましょう、36HとFAHとのXORの論理演算 をしてみましょう。

演算結果がCCHとなりました。この命令は、任意のビットを反転したりする ときなどに使われます。

表3-6 36HとFAHのXOR

ビット数	(36H)	(36H) (FAH)			
	A	?	В	=	Х
7	0	?	- 1	=	1
6	0	?	1	=	1
5	1	?	1	=	0
4	1	?	1	=	0
3	0	?	1	=	1
2	1	?	- 0	=	1
1	1	?	1	=	11
0	0	?	0	=	0

■ 3 つの XOR 命令

XOR 命令には、次の3種類があります。

XOR r

■ 式 XOR r (rはA, B, C, D, E, H, Lレジスタ)

■ 能 Aレジスタと「で指定されたレジスタとで、各ビットごとに XOR演算を行い、その結果をAレジスタにセットします。

例 XOR B

Aレジスタのデータ……1010011B

演算結果......00111000B

となります.

フラグ Sフラグ:命令を実行した結果が の変化

最上位のビット (7ビット目) が1なら1

最上位のビット (7ビット目) が0なら0

となります.

2フラグ:命令を実行した結果が

00Hのとき1

それ以外のときI

となります.

P/V フラグ:命令を実行した結果。1 になっているビットの数が

偶数なら1

奇数なら D となります

C フラグ: 常に 0 となります

XOR n

書 式 XOR n (n=0~255)

■ 能 Aレジスタと n データとで各ピットごとに XOR 演算を行い、その結果を Aレジスタにセットします。

例 XOR 78H

A レジスタのデータに A3H がセットされているとし、n のデータが 78H とすると、

Aレジスタのデータ·······10100011B nのデータ·········01111000B

演算結果-----11011011B

となります。

フラグ Sフラグ:命令を実行した結果が

の変化 最上位のビット (7ビット目) が 1 なら ! 最上位のビット (7ビット目) が 1 なら !

となります

2 フラグ:命令を実行した結果が

00日のとき1

となります

P/V フラグ:命令を実行した結果。 1 になっているピットの■が

偶数なら1 奇数なら0

となります。

C フラグ:常に D となります。

XOR (HL)

■ 式 XOR (HL)

能 AレジスタとHLペアレジスタが指定する番地のデータとで各

ビットごとに XOR 演算を行い、その結果を A レジスタにセットします。

(A) XOR (HL)

A レジスタのデータに A 3 H がセットされているとし、HL ベア レジスタが指定する番地のデータが A3H とすると

Aレジスタのデータ……10100011B

演算結果-----10011001B

となります。

§3 論理演算命令を使ってプログラムを作る

フラグ Sフラグ:命令を実行した結果が

の変化 最上位のピット (7ピット目) が1なら1

最上位のビット (7ビット目) が D なら B

となります。 2フラグ:命令を実行した結果が

00Hのとき1

それ以外のとき0

となります

P/V フラグ:命令を実行した結果、1になっているピットのIII

が 偶数なら1

奇数なら0

となります。 Cフラグ: 常に O となります。

131

1ビットごとの論理演算でドット単位の移動が可能に _____

論理演算の特徴は何と言っても、ビット単位の演算でしょう。つまり、1ビ ット単位の情報を扱うのです 使用範囲は結構広く。たとえばグラフィックス

に使います。PC-8801mkIISR のグラフィックス画面では、VRAM は1ビット が1ドットに対応しています。したがって、1ビットごとの論理演算を用いれ ば、ドット単位で画面を動かすことも可能になります。論理命令は是非とも覚 えなければならない命令なのです

■任意のビットを取り出したり消したり~AND

この命令は、特定のビットを取り出したいときや、特定のビットを消したい ときに使用されます。もし、任意のピットだけを取り出したいのなら、取り出 したいピットを1にして、この命令を実行すれば良いのです。1にしたビット だけを取り出せます。具体例を図3-5に示します。

図3-5 ANDを使って任意のビットをとり出す



AND演算を行うと



海算の結果…… 0 0 1 0 0 0 0

この場合は取り出すデータの5ビット目が1なので、演算の結果5ビ ット目が1になった。もし、取り出すデータの5ビット目が0ならば。 演算の結果5ビット目は0となる。

この AND 命令によるビットの取り出しは、他のビットに関係していません ので、この方法とは反対に、任意のビットだけを消すこともできます。つまり、 消したいビットを 0 にして他のビットは1にします。すると、 0 にしたビット だけを消すことができます。

■任意のビットを I にする~OR

この命令の目的は、任意のセットを1にすることです。この点は AND 命令と は遊です。AND 命令のときのようにセットしたいビットを1にして、OR 命令 を実行すれば良いのです。このOR命令では、任意のデータを消すことはでき ません。この命令の実行のやり方は AND 命令と同じです。

■"XOR OFFH"でもとのデータを反転

この命令は比較的使用されることの少ない命令です。"XOR 0FFH" で使用すると、もとのデータが反転されます。

論理演算の特殊な計算

歳理演算の特殊な計算を説明しましょう。それは次の3命令で、最も多く出てくるものです。

AND A

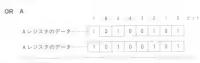
OR A

XOR A

つまりAレジスタ同士の論理演算です、「レジスタ同士で論理演算をやって何 になるのだろう」と思う人もいるかもしれませんが、ちゃんとある目的を果た してくれるのです。

■ Cy フラグをリセットする "AND A", "OR A"

まず, "AND A", "OR A" を実行してみます (図3-6).



演算の結果------1 0 1 1 0 1 0 1 は 結果は元のデータと同じ。

どちらの命令の結果も、もとのデータと同じになってしまいました。結果は 同じでも、しっかり計算をやってくれるのです。

何をするかと言うと、Cy フラグのリセットです。この命令を実行すると Cy フラグをリセットしてくれるのです。つまり。

Cy フラグをリセットするために、"AND A"、"OR A" を使用する

となります。

■データが00H かどうか調べるときにも使える

もうひとつは、2フラグの利用です。たとえば、あるメモリのデータが00Hであるかどうかを調べるときに "AND A"、"OR A" は利用されます。

まず、LD 命令などで調べたいメモリのデータをAレジスタにセットします。 その後で、"AND A"、"OR A" のどちらかの命令を実行します。もし、実行 結果が 0 であれば、 Z フラグは1となります。00日以外のデータであれば、 Z フラグは0となります。

■ "XOR A" でAレジスタをクリア

次に "XOR A" 命令を見てみましょう。 図3-7を見てください。

このように、すべてのビットはりになります。つまり、

Aレジスタをクリア (DDH) するために、"XOR A"を使用する

となります。Aレジスタをクリアする方法は"LD A, 00H"などがありますが、
"XOR A"は1パイトで良いので、1パイト分得します。ただし、この命令は
フララグやCv フラグをリセットしてしまうので、注意が必要です

キー入力のテストプログラムを作る

今まで長々と論理演算命令を説明してきましたが、グラフィックスの章に人 る前に練習をしておこうと考えたからです。

話をもとに戻して、論理演算命令を使用してリスト3-1を作ってみましょう。 CP 命令を使用していたのを、AND 命令に変えるだけです。

スペース・キーが押されたときのデータは図3-8のようになります。

AND 命令でZフラグが1になることを利用します。つまり、全部のビットの データを0にするのです。全部のビットを消すには、6ビット目を1にして他 のビットを0にします(図3-9)。

	7	6	5	4	3	2	1	0	Eo F
スペース・キーのデータ	1	0	1	1	1	1	1	1	(BFH)
AND 命令で用意するデータ	0	1	0	0	D	0	0	0	(40H)
演算の結果・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	0	0	0			0	0	D	

プログラムはリスト3-2のようになります。

■リスト3-2 (事業) 演算を使ったキー入力テスト (事業)

: ****** KEY IN TEST ********** :PROGRAM NAME --->kevt2.e : # = " - 019 / F2 | 7" | 12" O/ | AND J8H | ************************ MON: EQU 8E626H 8900 START: EQU 08900H ORG START B900 CD8B42 CALL 428BH A. (89H): I/O **-F / 89H 75 7"-9 7 32FN B983 III. MAIN: IN 40日 :スペース キーカーオサレタデータト イッチ スルカ ヒカク ス B985 E648 AND 8987 CARRES JP B90A 0E02 C.82H (ジ*カン マチ ノ ルーチン 890C 06FF LOP2: B. ØFFH 890E 05 LOP1: B. B90F C20EB9 NZ.LOP1 B912 0D B913 C20CB9 JP NZ, LOP2 B916 C303B9 JP MAIN 8919 END

キー入力とキャラクタの 移動プログラムを作る

キー入力とキャラクタの4方向移動プログラムを作る

キーボードからの入力原理が理解できたところで、実際のゲームプログラム などで使用されている。キー入力とキャラクタの移動プログラムを作ってみま しょう。

比較的良く使用されているのが、4方向の移動ですね。この原理さえ理解できれば、キー入力で頭を悩ますことがなくなります。

■移動に使う4つのキー

この4方向のキー入力でよく利用されるキーは、右側にあるテンキー(数値 入力用キー)です。 II 方向のキー入力には、図3-10のようなキーが利用されて います。



■面面の■』

次の両面の構成はプログラムを簡単にするために、図3-11のようなものとします。画面の隣にまたら■を画面からはみ出さないようにしなければなりません。そこで、警告のために、画面の隣にきたら■を赤に至えます。

13-11 画面の無漏



■インデックス・レジスタを使う

説用レジスタの中でボインタ(メモリなどを開接的。直接的に指定する役割) として他用できるのは、日Lペアレジスタしかありません、しかし、2-80ではこれを補うためにポインタ専用のインデックス・レジスタ (IX, IY) を持っているのです。このレジスタのおかげで、アドレッシングが非常に楽しなります。

インデックス・レジスタ (IX, IY) の使い方は HL ベアレジスタとほとんど 同じですが、IX, IY は16ビットですので、8 ビットずつ分けで使うことはでき ません。また、残念なことに16ビットの減算命合がありません。

インデックス・レジスタ (IX, IY) は、HLベアレジスタにない便利なアドレッシングを持っています。レジスタのデータの値を覚えずに±128の範囲で、 アドレッシングをすることができるのです。つまり、オフセットしたことと同じになります。

LD (IX+d), A

これはオフセット値を示します。

■ピットの符号付3進数を示しましょう。
8ビットの符号付2進数の例

+ 方向

0	1	2	3	 125	126	127
n	1	2	3	 70	7E	7F

§4 キー入力とキャラククタの移動プログラムを作る

一方向

-1	- 2	-3	- 4	126	- 127	- 128
FF	FE	FD	FC	82	81	80

例.

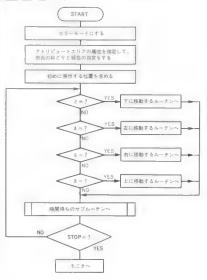
- LD IX, OF418H + インデックス・レジスタのIXにF418Hをセットします.
 - LD (IX+1), A ← F418H+1HしたアドレスのF419HにAレジスタのデータをセットします。
 - LD (IX+FEH), A ← F418H-02HしたアドレスのF416HにAレジスタのデータをセットします。

このアドレッシングは、レジスタのデータを変えないので、VRAMのアドレス指定に傾用です。この節のサンブルプログラムでも、アトリビュートエリアの城性の指定に使用しています。IY インデックス・レジスタは、IX インデックス・レジスタとなく同じです。

■プログラムのフローチャートを作る

ラフ・フローチャートは図3-12のようになります。まず、各レジスタの用途 は次の通りです。

図3-12 ラフ・フローチャート



§4 キー入力とキャラククタの移動プログラムを作る

HI ペアレジスタ・・・・・・・・VRAM のポインタ

IX インデックス・レジスタ······アトリビュートエリアの属性エリアのポイ ンタ

Dレジスタ………キャラクタの■の桁のカウンタ

■ HL ペアレジスタのデータとキャラクタの■■の■●

キャラクタのアドレッシングは HL ペアレジスタで行っており、キャラクタの移動と HL ペアレジスタとの関係は図3-13のようになります。



また、D、Eレジスタは移動した数を振えており、画面からはみ出さないようにしています

■アトリビュートの属性

アトリビュートの属性は、図3-14のようにしなくてはなりません。

	0		79桁
1行目		赤	
2行目	赤	緑	赤
行目	赤	緑	赤
行目		赤	

このように、2行目から23行目までは属性を3回指定しなければなりません。 1 行について 6 バイトのデータを書き込むことになりますが、この際、IX イン デックス・レジスタが傾利なのです

■ Cy フラグのリセット SCF, CCF 命令

Cv フラグをリセットするのに "AND A" 命令が使用できますが、別の命令 を健用してみましょう。次の2命令を組み合わせて使用します。

SCF……Cv フラグを端制的に1にする CCF·······Cv フラグを反転する

SCF, CCFの2命令を用いてCyフラグをリセットすれば、他のフラグはその ままになっています。したがって、他のフラグを壊したくないときに使用しま す また、価値演算のときにも使われています。

■キャラクタ4方向移動プログラム

特に難しくはないので、よく理解してください、なお、リスト3-4のプログラ ムではこれを応用して、キャラクタを8方向に移動させています。

■リスト3-3 キー入力とキャラクタ4方向移動プログラム



§4 キー入力とキャラククタの移動プログラムを作る

```
B900 011950 START: LD BC.5019H
 B983 3E81
                       LD A.01H
LD (8E6B2H),A
 B985 32B2E6
8995 328225 LD (825824).A

8987 3819 LD A.19H

899A 328356 LD (825831).A

8990 32635 LD (825831).A

8990 32626 LD (825831).A

8997 326426 LD (825841).A

8914 228826 LD (825841).A

8914 328826 LD (825881).A

8913 328926 LD (825891).A
                         79 1"1 7 Ph = 26 -----
: n-VA 7 72 &-F> # 37" ------
                         CALL 428BH
 B958 CD8B42
                         MAIN -----
 695B 118080 LD DE.8989H : DLジ 2タ コ ノクタ ,ELジ 2タ 9月 ノ 4 つの
895E 21C8F3 LD HL.8F3C8H : CRT ノ ヒタ リ ウェ ノ 4月
8983 23 LDM: INC HL

        8953
        Z3
        LOPM:
        INC
        HL

        8964
        14
        INC
        D

        8965
        3D
        DEC
        A

        8966
        C253B9
        JP
        NZ, LOPM

        8969
        3E0D
        LD
        A 40DH

        8968
        817808
        LD
        BC, 0878H

 B96E 09 LOPN: ADD HL.BC
B970 3D DEC A
B971 C26EB9
                                 JP NZ, LOPN
              :

EC LD A.CHR : キャラクタ〜 ヲ マンナカ ニ ヒョウシ スル

(D (NL.).A
 B974 3EEC
B976 77
8977 DB80 LOOPK: 1N A.(80H) :2 カーオサレタカ シラヘール
B979 47 E684 AND 84H
B97C CASPBS JP Z.DOWN
```

■3章 キーボードを操作す

	;			
B97F 78 B980 E610 B982 CAB2B9				:4 カー オサレタカ シラペール
8985 78 8986 E640	L. A	D ND	A.B 40H Z.RIGHT	16 ກື ጳ ሣレダカ シラヘ [™] ル
B98B D801 B98D E601 B98F CAD2B9	I A J	N ND IP	A.(81H) 81H Z.UP	:8 ከ" ጳህኒያክ ይማንግል
B992 CDE8B9	LOPE: C	ALL	WAIT	
B995 DB@9 B997 E601 B999 C277B9	I A J	N ND	A.(89H) 01H NZ.LOOPK	;STOP ቀ- ከ" オワレタカ シラヘ"!!
B99C C326E6	J	p	MON	: E=9- \ E+*IL
	1074	-5 8	- チン ・・・・	
899F 7B 89A0 FE18 89A2 CA92B9	DOWN: L	D P	A.E 18H Z.LOPE	こシタ へ イトニウ ズル ルーチン
B9A5 AF B9A6 77	X	OR D	A (HL),A BC 00784	;A レジ*スタ ラ クリア スル
B9AA 69 B9AB 3EEC B9AD 77	A L	.D .D	HL,BC A,CHR (HL),A	
B9AE 1C B9AF C392B9	J	NC IP	E LOPE	
B9B2 7A B9B3 FE00 B9B5 CA92B9 B9B8 AF B9B9 77 B9B8 2B B9B8 3EEC B9BD 77 B9BE 15 B9BF C392B9	LEFT: L C J X L D L L D	D P OR .D D D D D D D D D D D D D D D D D D D	A.D ØØH Z.LOPE A (HL),A HL A.CHR (HL).A D LOPE	:59 1 4110 XL L-F5 :(A L5129 9 997 XL :(E919 1 4110 XL L-F5)
B9C3 FE4F	C	P	4FH	: さず゛ヘ イト゛ウ スル ルーチン
BSC5 CA92B9 BSC8 AF BSC9 77 BSCA 23 BSCB 3EEC BSCD 77 BSCE 14 BSCF C392B9 BSD2 7B BSD3 FE00 BSD5 CA92B9 BSD5 CA92B9	L L J	D D NC	A.CHR (HL),A D LOPE	
89D2 7B B9D3 FE00 B9D5 CA92B9 B9D8 AF	ÚP: L G J X	D P IP OR	A.E ØØH Z.LOPE A	:ウェ へ イト [*] ウ スル ルーチン

§4 キー入力とキャラククタの移動プログラムを作る

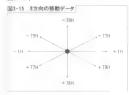
B9D9	77		LD	(HL),A		
RODA	017800		LD	BC,0078H		
BODD	37		SCF	DC (DD / OIL	CALL TO THE WAY OF THE PARTY.	
					;キャリーフラクト ヲ ! ニスル	
BADE	3F		CCF		;キャリーフラク [*] ヲ W ニスル	
B9DF	ED42		5BC	HL, BC	: (キャリーフラク* E ヒク I	
B9E1	SEEC		LD	A.CHR		
	77		LD	(HL), A		
89£4						
			DEC	E		
B9E5	C392B9		JP	LOPE		
		15" 70	24 /	Int		
		12 00	111	~ / /		
B9E8	0E0F	WAIT:		C.0FH		
B9EA	0 SFF	WLP2:	L.D	B. WFFH		
BREC	0.5	WLP1:	DEC	P.		
BSED	C2ECB9	W-23 1	JP	NZ. WLPI		
BBF0	6 D		DEC	C		
B9F1	C2EAB9		JP	NZ.WLP2		
B9F4	C9		RET			
B9F5			END			
Data			END			

8方向の移動プログラムを作る

前のプログラムを少し複雑にして、8方向に移動するプログラムにしてみま しょう。プログラムはほとんどそのままで、斜めに移動するルーチンを追加し ただけです。

■斜めに移動するには

斜めに移動する方法は簡単で、HL ペアレジスタに加減するデータ78日を1 プラス、マイナスするだけです (図3-15)



第3章 キーボードを操作する

■使用するキー

使用するキーは、1、2、3、4、6、7、8、9のキーです。5のキーは 使用しません(■3-16)。



■リスト3-4 キー入力とキャラクタ#方向移動プログラム : **** COLOR TEXT MODE **** :PROGRAM NAME ---> keyin8.e * キーニュウリョク デ^{*}モ フ^{*} グ^{*} ラム 2(8 本ウコウ) :キャラクター ノ イトこう キー 7 | 9 STOP #- 7" E=9- 1 :PROGRAM STRAT ADDRESS START: EQU @B988H (モニター / エントリ お イント B900 MON: EQU #E626H CRT コントロール ルーチン COLOR: EQU 6F68H ; *+509- / コート` CHR: EQU ØECH SSEC ORG START CRT COLOR MODE

§4 キー入力とキャラククタの移動プログラムを作る

```
8980 011950 START: LD BC.5019H
    B989 811958 START: LD BC.5819H
B989 3584 LD A.81H
B985 3232E6 LD (BEDEZH).A B988 3232E6
LD A.1998 3219 LD A.1998 3219 LD GEDEZH).A B984 A.1998 3219 LD GEDEZH).A B986 A.1998 A.19
                                                                                                                                                                        : 72 FT 9 Fh = 2% -----
    :カーソル ヲ ケス ルーデン ヲ ヨフ -----
                                                                                                                                                                                        CALL 428BH
            B958 CD8B42
### 10 DEC A | 1 No. |
            LD (HL), A (188H) : ארייני מעופא ארייני מער אייני מער ארייני מער ארייני מער ארייני מער ארייני מער ארייני מער אייני מער ארייני מער אייני מער איי
```

第3章 キーボードを操作する

B97C	CAB8B9		JP	Z,LDOWN	
	78 E604 CAD2B9		AND	A - B 8 4H Z - DOWN	;2 か オサレタカ シラヘニル
	78 E608 CAE5B9		AND	A.B 88H Z.RDOWN	:3 か" オサレタカ シラヘ"ル
	78 E610 CAFFB9		LD AND JP	10H	;4 か" オサレタカ シラヘ"品
	78 E640 CA0FBA			A.B 40H Z.RIGHT	:6 か* オサレタか シラヘ*ル
	78 E686 CAIFBA			A.B 80H Z.LUP	:7 か オサレタカ シラヘーロ
			IN LD AND JP	A,(81H) B,A 81H Z,UP	:8 か [*] オサレタカ シラヘ [*] ル
	78 E602 CA52BA		AND	A.B 02H Z.RUP	:9 ก้ สซีป <i>ซี</i> ก อวิจานี
B9AB	CDSFBA	LOPE:	CALL	WAIT	
8980	DB09 E601 C277B9	;	AND		:STOP #- ガ オサレタカ シラベル
B9B5	C326E6		JP	MON	: E29- \ EF*A
		/	(F*5)	l-7>	
B9B9 B9BB B9BE B9BF B9C1 B9C4 B9C5 B9C6 B9C6 B9C0 B9CD B9CD B9CE	FE18 CAABBS 7A FE00 CAABBS AF 77 017780 89 3EEC 77		CP JP LD CP JP XOR LD LD LD LD LD LD LD LD LD LD LD LD LD	18H 2.LOPE A.D 89H Z.LOPE A (HL).A BC.8877H HL.BC A.CHR (HL).A	<u>:</u> የድምህይቃ ላ ፈትግሳ
B9D2 B9D3		DOWN:		A.E 18H	(39 1 4179

§4 キー入力とキャラククタの移動プログラムを作る

DODE CAADO	10	10	7. 1. 11111	
89D5 CAABB B9D8 AF B9D9 77 B9DA 01780 B9DD 09 B9DE 3EEC B9E0 77 B9E1 1C B9E2 C3ABB	9	JP XOR LD LD LD LD LD INC JP	Z.LOPE A (HL).A BC.8878H HL.BC A.CHR (HL).A E LOPE	
B9E5 7B B9E6 FE18 B9E8 CAABB B9E8 7A B9EC FE4F B9EC AASB B9F1 AF B9F2 77 B9F6 89 B9F3 81798 B9F9 77 B9FA IC B9FB 14 B9FC C3ABB	9	LD CP JP XOR LD LD ADD LD LD INC JP	A,E 18H Z,LOPE A,D 4FH Z,LOPE A (HL),A BC,0079H HL,BC A,CHR (HL),A E D	: おキトシタ へ イト・ウ
B9FF 7A BA88 FE88 BA82 CAABB BA85 AF BA86 77 BA87 2B BA88 3EEC BA8A 77 BA8B 15 BA8C C3ABB		LD JP XOR LD DEC LD LD DEC	A,D 88H Z,LOPE A (HL).A HL A,CHR (RL).A D LOPE	: ೬ ቃግሀ ∿ ፈ ኑ ግዕ
BA8F 7A BA18 FE4F BA12 CAABB BA15 AF BA16 77 BA17 23 BA18 BEEC BA1A 77 BA18 14 BA1C C3ABB	9	LD JP XOR LD INC LD LD LD LD LD LD LD LD LD LD LD LD LD	A.D 4FH Z.LOPE A (HL).A HL A.CHR (HL).A D LOPE	፡ ፰ቱ° ヘ ፈኑ° ゥ
BA1F 7B BA26 FE96 BA22 CAABB: BA25 7A BA26 FE96 BA28 CAABB: BA2B AF BA2C 77 BA2D 81798! BA39 37 BA31 3F BA32 ED42 BA34 AFCC	9		A.E 88H Z.LOPE A.D 88H Z.LUPE A (HL).A BC.8879H	(ዚያካያዕ፤ ላ ብዮዕ

第3章 キーボードを操作する

```
BA36 77
                     LD
                          (HL).A
                     DEC
BA37 1D
                         E
BA38 15
                     JP
                         LOPE
BA39 C3ABB9
BA3C 7B
             UP:
                                            101 A 4170
                         A.E
BA3D FE00
                         ABH
                     JP
BASF CAABBS
BA42 AF
                     XOR
                         A
BA43 -77
                         (HL),A
                          BC,8078H
BA44 81780B
BA47 37
BA48 3F
BA49 ED42
                     SBC
                         HL, BC
BA4B 3EEC
BA4D 77
                    LD
                         (HL),A
BA4E 1D
                     DEC
BA4F C3ABB9
                     JP
                         1.0PE
BA52 7B
              RUP:
                   LD
                        A.E
                                            134701 5 4170
BA53 FE00
                        86H
                     JP
BASS CAABBS
                     LD
BASS FE4F
                         4FH
                     JP
BASB CAABBS
                     XOR A
BASE AF
BASE 77
                     LD
                         (HL),A
                          BC.8077H
BAS8 817708
BA63 37
                     SCF
BA64 3F
BA65 ED42
                     SBC HL, BC
BA67 SEEC
                     LD
                          A. CHR
                     LD
                          (HL),A
BAGA ID
                     DEC
BA6B 14
                     INC D
BASC C3ABB9
                     JP LOPE
              こうこカン マチ ノ ルーチン ----
BASF BEBF
              WAIT:
                    L,D
                         C. BFH
BA71 06FF
              WLP2: LD
                         B. ØFFH
BA73 85
              WLP1:
                     DEC
                         - 8
BA74 C273BA
                     JP
                         NZ., WLP1
BA77 BD
                     JP
BA78 C271BA
                          NZ. WLP2
BA7B C9
BA7C
                   END
```

絵描きプログラムを作る

ちょっと応用すると、リスト3-4は絵描きのプログラムになります。リスト3 -4でキャラクタの移動の際に消しているのを、消さないでそのままにしている

だけです。

●リスト3-5 絵描きプログラム

```
:**** COLOR TEXT MODE ****
             :PROGRAM NAME ----> kevinP.e
             ; キーニュウリョク デキモ ファクトラム 3(エカキ )
             (キャラクター ノ イト・ウ キー
                           7 8 9
                           4-6
                                         CLS #- 7" 29724
                                         STOP +- 7" t=9~ 1
            6F6B
RREC
             CHR: EQU
                        RECH
                    ORG START
              CRT COLOR MODE -----
             START: LD BC.5819H
B988 811958
             LD A,01H
LD (0E6B2H),A
LD A.19H
B903 3E01
8985 32B2E6
B988 3E19
                   LD
B96A 32B3E6
                        (@E6B3H),A
898D 3EE8
                    LD A. ØE8H
B90F 32B4E6
                   LD
LD
                        (@E6B4H).A
                        A.88H
B912 3E00
B914 3288E6
                   LD
                        (BEGBSH), A
                   LD A. BFFH
LD (BE6B9H).A
8917 3EFF
B919 3289E6
                    CALL COLOR
              : 72 FT 7 Fh = 20 -----
;ウェ ノ カラー
;アカ
8930 DD368980 LD (IX+0)・68日 :セターリッシ / カラー
8936 DD368148 LD (IX+1)・48日 :アカ
8936 DD368201 LD (IX+2)・61日 :オュウルン / カラー
8937 DD368281 LD (IX+2)・61日 :オュウルン / カラー
B936 DD360148 LD (1X+1).48H
B93A DD360201 LD (1X+2).01H
B93E DD360388 LD (1X+3).088H
                                            :キャラクター ノ カラー ミトニリ
```

第3章 キーボードを操作する

8946 894A 894B	DD36044F DD360548 05 C230B9 DD19		LD LD DEC JP ADD	([X+4),4FI ([X+5),48I B NZ,LOPI 1X,DE	H		: ± : 7	キ [*] ハシ ノ カラー カ	
B958 B954	DD368688 DD368148		LD	1X.DE ([X+0),001 ([X+1),481	H		: 2°	タノ カラー カ	
		トカーソル	ラ ケス	M-+> ∃ ∃7°					
B958	CD8B42		CALL	428BH					
		MAIN							
B95E B961 B963 B964 B965 B966	14 3D C263B9	LOPM:	LD INC INC DEC JP	HL.8F3C8H A.27H HL D A NZ,LOPM	; D	L≎™ RT	スタ ヨコ ノニヒタ [™] り	ノ サタ 、Eレジ・スタ タ ウェ ノ イチ	7 / ∜°∃Ó
B968 B96E B96F B978	3E8D 617888 69 1C 3D C25EB9	LOPN:	ADD 1NC DEC						
	3EEC 77		LD	A.CHR (HL),A	;	†+7.	79- F	マンナカ ニ とョウシ゛ ス	t
B979 B97A	DB00 47 E602 CABFB9	LOOPK:	AND	A.(00H) B.A 02H Z.LDOWN	; [ħ*	オサレシカ	591°E	
B97F B988 B982	78 E684 CAD9B9		LD AND JP	A.B 04H Z.DOWN	;2	カ゛	オサレタカ	5911	
B986	78 E608 CAECB9		AND	A.B 88H Z.RDOWN	:3	かっ	オサレタカ	シラヘト目	
	78 E618 CA06BA		AND	A,B 10H Z.LEFT	; 4	21.	オサレタカ	3 う へ"ル	
B992	78 E648 CA168A		AND	A.B 40H Z.RIGHT	: 6	<i>h</i> ~	オサレタカ	594° l	
B997 B998 B99A	78 E688 CA26BA		AND	A,B 80H Z.LUP	:7	'n"	オサレタカ	シラハール	
B99F	DB01 47 E601		IN LD AND	A.(81H) B.A 81H	: 8	カー	すサレタカ	シラヘール	

§4 キー入力とキャラククタの移動プログラムを作る

B9A2	CA43BA		JP	Z.UP	
B9A6	E682		AND	0·2H	וירפ מפלודה אין פּוּן פּוּ
BSAB	CD96BA		CALL	WAIT	
B986	E601		AND	919	:CLR +- n~ オサレタカ シラハー&
B9B7	E691		AND	01H	;STOP ÷- ກ"
вявс	C326E6		JP	MDN	: モニター へ モトニル
			(1-7)	N-50	
B9C0	FE18		CP	1.8H	: E97 959 A 4F76
89C5 89C6	7A FE88		LD	A.D 80H	
B9CB B9CC	8 9 8 8		NOP		
8900 8901	SEEC SEEC		ADD	HL.BC A.CHR	
B9D4 B9D5	1 C 1 5		INC	E D	
B9DA B9DC	FE18 CAABB9		CP	18H	;ይቃ ∿ ብ⊱*ጛ
89E1 89E4 89E5	817888 3EEC		ADD	HL.BC A.CHR	
B9E8	1 C		1NC	E	
89ED 89EF 89F2 89F3 89F8 89F8	FE18 CAABB9 7A FE4F CAABB9 88		CP JP LD CP JP NOP NOP	Z.LOPE A.D 4FH Z.LOPE	: ミキ*シタ へ イト*ウ
B9FD	89				
	B9AS B9AE B9AE B9BC B9BC B9C B9C B9C B9BC B9BC B9BC B	B9D9 78 B9D4 FE18 B9D6 CAABB9 B9D7 B9E8 89 B9E1 817888 B9E2 817888 B9E7 77 B9E8 1C B9E9 C3ABBS B9EC 78 B9EC 78 B9EC 78	BASA CD96BA LOPE: B9A6 CB92 B9A6 CA99BA LOPE: B9A6 CB9A6 CB	### B904 CAS9EA LOPE: CALL ### B905 DB89 CAT6BA LOPE: CALL ### B905 DB89 CACABB9 LOPE: CALBB9 CALBB9 CACABB9 LOPE: CALBB9 CACABB9 LOPE: CALBB9 C	BASE CASES CALL WAIT

第3章 キーボードを操作する

BARR BARR BARR BARR	1 C		INC INC JP	(HL),A E D LOPE	
BA07 BA09 BA0C BA0D BA0E BA0F BA11 BA12	89 2B 3EEC 77		JP NOP NOP DEC LD	Z.LOPÉ	፤ ቲ ያግሀ ጎ ፈዩጉዕ
BA17 BA19 BA1C BA1D BA1E BA1F BA21 BA22	00 23 3EEC 77	RIGHT:	LD CP JP NOP NOP INC LD LD LD	A,D 4FH Z,LOPE HL A,CHR (HL),A	(24* ^ 4)** 7
BA27 BA29 BA2C BA2D BA32F BA33 BA34 BA37 BA38 BA39 BA3B BA3B BA3F	FE00 CAABB9 00 017900 37 3F ED42 3EEC 77	LUP:	CP JP LD CP JP NOP LD SCF CCF SBC LD LD LD LD DEC	BC.8079H HL.BC A.CHR (HL).A	נפירפים: א בפיניפים:
BA44 BA46 BA49 BA4A BA4E BA4F BA50 BA52 BA54 BA55	88 817888 37 3F ED42 3EEC 77		JP NOP LD SCF CCF SBC LD LD DEC	A.E 80H Z.LOPE BC.0878H HL.BC A.CHR (HL).A E LOPE	:01 0 410
BA59	7B	RUP:	LD	A,E	こうきつりょ へ イトつり

§4 キー入力とキャラククタの移動プログラムを作る

```
BASA FE00
                          BBH
                      JP
BASC CAABBS
BASE 7A
                      LD
                           4FH
BA60 FE4F
                      JP
                           Z.LOPE
BAG2 CAABB9
BA65 88
BAGG III
BA67 817789
                      LD
                           BC,0077H
BAGB 3F
BAGC ED42
                      SBC
                           HL.BC
BAGE SEEC
                           A.CHR
                           (HL).A
                      DEC
BA71 10
                           E
                      INC
BA72 14
                           LOPE
                      JP
ВА73 СЗАВВ9
                                                :CRT / 297
BA76 D5
                      PUSH DE
                           IY. BF3C8H
BA77 FD21C8F3
                      LD
                      LD
BA7B 112800
BATE AF
                           C. 19H
BA7F 8E19
                           B.58H
                      LD
 BA81 8658
BA83 FD7708
                      LD
                      INC
BASS FD23
                       DEC
 BA88 85
                      JP
                           NZ.CLLP1
 BA89 C289BA
 BASC FD19
                           1Y.DE
 BASE 8D
                      JP
                            NZ.CLLP2
                           DE
 BA92 III
                      JP
                           START
 BA93 C300B9
                こら カン マチ ノ ルーチン ーーーーー
               WAIT:
                      ŁD
                           C, 8FH
 BASS ØERF
BASS GEFF
                      LD
                           B. ØFFH
               WLP2:
               WLP1:
                      DEC
                            В
 BASA 85
 BASB C29ABA
                       JP
                            NZ. WLP1
                       DEC
 BASE BD
                            NZ, WLP2
 BASE C298BA
                       JP
                       RET
BAA2 C9
 BAA3
```

これでキー人力の説明を終りますが、理解できたでしょうか? この章を読み終えると楽なものですね。次章はグラフィックスの世界に入ります。





4 グラフィックスの世界

グラフィックス・パターンの作り方と■■方法

今のパソコンには、グラフィックス機能がついているのが当たり削になっています。グラフィックス機能は美しいのでパソコンの魅力のひとつとなっていますが、BASIC 命令で使用すると違くてイライラします。BASIC 命令だけで、グラフィックスを利用したリアルタイム・ゲームを作るのは事実上不可能で、当然、スピードの速いマシン語を使わなければなりません。

せん。 そこでこの章と次の章で、皆さんをグ ラフィックスの世界に招待します。この 章では何ビットかの集よりを1ブロック として、ブロック単位でパターン(ヘリ コブター)を移動させてみましょう。

グラフィックスの原理

ビット・マップ法によるグラフィックス

TV (CRT を含む) に描き出される画像は、点の基まりから出来ていること は知っていると思います。この画像を作る点を画着(ドット)といいます。こ の画表が多いほど鮮明な画像を描くことができることになります。PC-8801 mkliSR では図4-1のようになっています。

このように、カラー両面ではモノクロ両面の半分の表示能力しかありませんが、ゲームなどに使用するには十分です。

■カラーの■■■成をくわしく見ると…

もう少しカラーの画面構成をくわしくみることにしましょう。カラー画面の を上の部分は図■2のようになっています。

図4-2 カラー画画の左上を(わしく見書)



このように、各ビットの表し方は8ビットを1つのブロックとして扱っています、つまり、1パイトを1つの単位としていることになります。このことは 単ビット・マシンに都合の良いことです。

■ | ドットが 1 ビットに対応するビット・マップ法

1ドットは、1ビットに対応しています。このことを、ビット・マップ法と 言います。

ビット・マップ法 → 1.ドットが1.ビットに対応している VRAM 構成 ビットが1.のとき………ドットが表示される ビットが1.のとき……...ドットが過去される

この方法はグラフィックス・パターンのデータを作るのに、比較的衆な方法です。高級タイプのパソコンはこの方法が多いようです。

したがって、実際のグラフィックスのパターンを作るときは、8 ビットごと に1パイトのデータに変換してやる必要があります。このグラフィックス・パ ターンのデータ変換は予作姿でやらなければなりません。

■カラーモードにおける G-VRAM の番地

8 ビットを1 単位としているのですから、この1 単位ごとに VRAM の番地 が割り振られています。横の番地数は、

640ドット÷ II ビット=80割地

となります。グラフィックスの VRAM 香地(G-VRAM)は図4-3のようになります。ただし、テキストモードのときのように、アトリビュートエリアはありません。

図4-3 カラーモード時のG-VRAMの書地

1	2					÷									79	80//1
C000	C001														C04E	C04F
C050	C051														COAE	COAF
		٠	•	•	٠	٠	•	٠	٠	•	٠	٠	•	•		
FDE0	FDE1			,											FE2E	FE2F
FE30	FE31														FE7E	FE7F

このように横80パイトで縦が200ラインで200パイトになりますから、 80パイト×200パイト=16000パイト(16Kパイト)

つまり、600ドット×200ドットで16Kバイトのメモリ空間が必要になるわけです。

グラフィックス・パターンをカラーにするには

グラフィックス・パターンをカラーにするには、少し複雑になります。グラフィックスでは、カラーテレビの原理をそのまま使用しています。カラーテレビは重(赤)、G(縁)、B(音)の3原色を微妙にコントロールしていろいろな色を出しています。

グラフィックスのカラーも原理は同じでR, G, Bに対応する VRAM を持っています。1つの色について16Kバイト必要ですから、カラーにするためには、

16Kバイト×3カラー分=48Kバイト

必要となります。

■バンク切り替えで48Kバイトの G-VRAM

48Kバイトもメモリを必要とするので、メイン・メモリのワークエリアが接 らないように、バンク切り替えという方法をとってメイン・メモリが減らない ように工人しています。このバンク切り替えという方法にZ-80 CPU を使用し ているパソコンでは、ごく当たり前のことです。このバンクの切り替えは "OUT"命令を使用することによって、ソフトウエア的に切り替えができます。 このカラー化のためのVRAMをブレーンとも呼びます。PC-8801mkIISRの ブレーンは個4-4のようになっています。

図4-4 SRØG-VRAM





■3プレーンで8色出せる

どのプレーンにグラフィックス・パターンを書き込むかによって、グラフィックス・パターンに色がつきます。3プレーンありますので、プレーンの組み合わせで8色まで出ます。プレーンと色の関係は散4-1のようになります。

もし、グラフィックス・パターンを白にするなら3プレーンにデータを書き 込まなくてはなりません。

■4-1 プレーンと■の関係

色	プレーン2	ブレーン1	プレーンⅡ
100	G	R	В
502 7077	0	0	0
额	0	0	1
赤	0	1	0
紫	0	1	1
緑	1	0	0
水色	1	0	1
黄	1	1	0
白	1	1	1

カラープレーンを切り替えるには

THE PROPERTY OF THE PROPERTY O

マシン語で直接 G-VRAM を操作する際、バンク切り替えに"OUT"命令を 使用するのは良いのですが、その際には次の細かい条件を守って使用しなけれ ばなりません。

G-VRAM 操作上の注意

- I. G-VRAM のアクセス・ルーチンは、必ず C000H 番地以前になくてはなりません。
- バンク切り替えの前に、必ず"DI"命令をしなくてはなりません。割り込みによって起こる暴走を防ぐためです。
- G-VRAM のアクセスが終ったならば、バンク切り替えでメイン・RAM に必ず切り替え、"EI" 命令を脈行しなければなりません。

この条件はどんなことがあっても必ず守ってください。もし、この条件のひ とつでも守らないと禁走を起こします。

■割り込みをコントロールする DI、EI 命令

PC-8801mkIISRのCRTコントローラは制り込み処理をしているので、ハードウェアで定期的にシステム・RAM(E600日番地以降)を読み取りにきます。 このときに、正しいデータでないと暴走を起こします。暴走しないように、シ

ステム・RAM 以外をアクセスしているときは割り込みが起こらないようにしなければなりません。割り込みをコントロールしているのが、この2つの命令なのです。

DI命令・・・・・割り込み禁止 EI命令・・・・・割り込み許可

■カラープレーン切り替えの手順

実際の手順は次のように行います。

DI OUT (GRAM), A:プレーンの I/O アドレス :: :プレーンの音を込みルーチン OUT (SFH), A :メイン・メモリにする EI ::

なお、このAレシスタのデータは意味のないものです。したがって、どんな データでもかまいません。

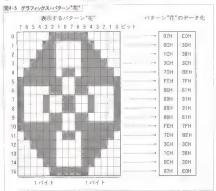
このようになりますが、実際のサンプル・プログラムをみれば理解できるで しょう。

グラフィックスの プログラムを作る

グラフィックス・パターン*花"の表示

グラフィックスの原理が理解できたところで、実際のサンプルプログラムを 作ってみましょう。

まず初めは、簡単なグラフィックス・パターンの表示からです。 🖫 -5のようなグラフィックス・パターンを表示してみましょう。



このままのパターンでも良いの ですが、パターンを移動したりす もときに、前のパターンの残りが 出ないようにするために、パター ンのまわりを00日で開んでおく と、プログラミングが繁になりま す、では、パターンをデータ化し てみましょう(間4-2)、

このように、並ぶしようとする グラフィックス・パターンは、す ベてデータ化しなければなりよ。す 人、複雑なグラフィックス・パタ ーンほど、データ化がやっかいで す、パターン "化"のデータ化の 手順は図4-60ようになります。 なお、わざわざ2進数の0、1

で表す必要はありません。直接、 16進数でデータ化してしまえば良いのです。

表4-2 プログラムトのパターンのデータ化

	. / -/	>====	- 4//	716
	00	-00	00	00
	00	07	E0	00
	00	OC	30	00
	00	1C	38	00
	00	3C	3C	00
	00	7D	BE	00
	00	FE .	7F	00
	00	86	61	00
	00	89	97	00
	00	89	91	00
	00	86	61	00
	00	FE	7F	00
	00	7.D	BE	00
	00	3C	3C	00
	00	1C :	38	00
	00	OC.	30	00
	00	07	EO	00
	00	00	00	00
_				

この00Hのデータは、パターンがどちらの方向に照 しても、前のパターンを消すのに必要。

	7	6	5	4	3	2	1	0	7	8	5	4	3	2	1	0
パターン 0																
# 遊散で表す	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	
16進数で表す			0				7			E)	

CRT中央に"花"を表示するプログラムの作成

それでは、CRT上の中央にバターン"花"を表示するプログラムを考えてみ

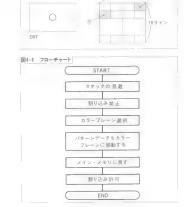
ましょう、パターン "作" は4パイト×18ラインですから、1ラインの4パイトをカラープレーンに青き込んだ接は、下のラインを表示しなければなりません。表が依拠を下のラインまで下げるには、現在の番地に50日 番地ゼします。 テキストモードでは78日 を足していましたが、グラフィックスモードではアトリビュートエリアがないので、80桁分の50日で長いのです。

実際には**図4-7**のとおり、A点からB点に行かなければならないので、4 パイト分引く必要があります。A点からB点に移動するときに加えるデータは、 DD5DH-0004H=004CH

となります.

フローチャートを作りましょう。図4-8のようになります。

图4-7



■スタック・エリアを広げる方法

グラフィックスのプログラムでは、スタック・ポインタをよく使用するので、 自分のプログラム用にスタック・エリアを広げておかなければなりません。た だし、現在のスタック・ポインタの保置を保存しておく必要があります。そう しないと暴走してしまいます。スタック・エリアを広げる方法を示します。

スタック・ポインタの設定

LD(SPSAVE)、SP……現在のスタック・ポインタ(SP)の位置をSPSAVEと 名のついたエリアに保存

LD SP, STACK………新たにSTACKの位置にSPを設定

SPSAVE: DS 02H……元スタックデータ

ル)のスタック。エリアを確保した

■"花"のプログラム

さて、プログラムは次のようになります。

●リスト4-1 "花"表示プログラム

: GRPH--1 : PROGRAM NAME ---> Grpb1.e *********************** B988 START: EQU @B980H ERRR GRAMS: EQU ØE068H MON: EQU BE626H GRAM8: EQU 5CH 005D GRAM1: EQU 5DH GRAM2: EQU 5 EH MRAM: EQU 5FH OPG START B988 E07374B9 LD (SPSAVE).SP : 5° ンサ° イ ノ スタック タイヒ B984 319EB9 LD SP. STACK : T59= 29=2 7 t=74

LD HL, GRAMS

BORT TIRRER

§2 グラフィックスのプログラムを作る

```
B90A 1129B9
                      LD
                             DE, CHRDATA
                      D1
B980 F3
                      OUT (GRAMI), A
890E D35D
               LD C.12H
LOP2: LD B.04H
LOP1: LD A.(DE)
B918 0E12
B912 0684
8914 1A
B915 77
                       LD
                            (HL) A
B916 23
B917 13
B918 05
                       INC HL
                      INC DE
                      DEC B
                   JP NZ.LOP1
PUSH BC
LD BC.004CH
ADD HL.BC
B919 C21489
B91C C5
891D 814C88
B920 89
B921 C1
                      POP BC
B922 ■
                      DEC
                      JP
                             NZ.LOP2
B923 C212B9
8926 D35F
                     OUT (MRAM), A
                      E1
B928 FB
                CHRDATADS 88
B929 B8
                DB
B92A 00
                           8.8
8928 III
                       DB
                           88
892C BB
                       DB
                           0.0
                           80
892D 88
               DB
892E 07
892F E0
8936 86
8931 BB
8932 BC
B933 30
B934 88
B935 86
B936 1C
B937 38
B938 80
B939 00
893A 3C
B93B 3C
B93C 00
B93D 88
B93E 7D
B93F 8E
                 D8 8BEH
DB 68
DB 89
DB 9FEH
DB 7FH
DB 69
DB 96
8948 88
8941 68
8942 FE
B943 7F
B944 86
B945 88
B946 86
                      DB
                            868
B947 61
                      DB
                           61H
                   DB
B948 86
8949 88
                      DB
                           67.81
                      DB
B94A 89
                           89H
B94B 91
                      DB
                             918
```

```
B94C 88
                       DB
                             88
                            68
894D 88
B94E 89
894F 91
                             88
8951 88
                             8B
B953 61
8954 KH
                             9.6
                            6.6
B957
                             7EH
                             9.8
R959 88
                             98
8958 BE
895C 86
                             98
8962 FW
                             0.0
                             9.6
B963 38
                             38H
8964 88
8965 00
                             88
8966 RC
                             SCH
                             36H
B968 III
8959 68
                            9.6
                       1908
B98C 88
                             88
896D 08
     88
     9.8
                       1000
                             80
                             BB
                            MON
                           0.2H
                                                   : FF / 2999 F'-9
899E
```

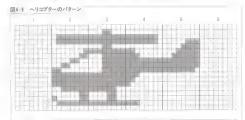
ヘリコプターを表示する

パターン"化"は16ドット×16ドットの正方形の表示だったのですが、CRT の関係上、縦技の表示パターンになってしまいました。もし、正方形型に表示 したいなら、橋のドット数を3 作にしてやらなければなりません。

ここで作るプログラムは、バターン"花"と全く同じ方法で作るものですが、

もう少し複雑にして、後のプログラムでも使用できるようなものにしてみましょう。ここでは、正方形型になるようにしてあります。

表示するのは関4-9のようなヘリコアターのバターンです.



ヘリコプターのバターン・データは表4-3のとおりです。

表4-3 ヘリコプターのパターン データ

OCH. DOH. GOH. DOH. OCH. COH. OOH, OOH, 18H, OOH, OOH, OOH 00H, 07FH, 0FFH, 0FEH, 00H, 00H 00H.07FH 0FFH.0FEH.07H.00H 00H, 00H, 18H, 00H, 0FH, 00H OOH, 01H, 0FFH, 0C0H, 1FH, 00H 00H, 03H, 1FH, 0E0H, 3FH, 00H OOH, OCH, 1FH, OFFH, OFFH, OOH 00H, 30H, 1FH, 0FFH, 0FFH, 00H OOH, OCOH, 1FH, OFFH, OFEH, OOK OOH, OFFH, OFFH, OEOH, OOH, OOH DOH, OFFH, OFFH, OCOH, OOH, OOH 00H, 0FFH, 0FFH, 80H, 00H, 00H, 00H, 3FH, 0FCH, 00H, 00H, 00H 00H, 03H, 0CH, 00H, 00H, 00H ODH. 83H. OCH. ODH. ODH. ODH. ODH, 7FH, 0FFH, 0E0H, 00H, 00H, OCH. DOH. DOH. OOH. OOH. OOH

プログラムはリスト4-2のようになります。

■リスト4-2 ヘリコプターを表示する

		,	****		**********
		GRPH-	-2		
		: PROGR	AM NAI	WE> Grph2.e	
		:			********
B900				0B908H 0E008H	
E000 E626 005C 005D 005E 005F		GRAMS: MON: GRAMB: GRAM1: GRAM2: MRAM:	EQU EQU EQU	9E626H 5CH 5DH 5EH 5FH	
			ORG	START	
	ED739889 31D6B9			(SPSAVE).SP SP.STACK	: ケーンサーイ ノ スタック ノ タイヒ : アラタニ スタック - ラ セッテイ
	2100E0 112CB9		LD LD	HL.GRAMS DE.CHRDATA	
B90D	F3		D1		
B90E	D35E		OUT	(GRAM2).A	
8912 8914 8915 8916 8917 8918 8919 8910 8920 8921 8922 8923 8923	77 23 13 85 C21489 C5 814480 89 C1 80 C21289	LOP2:	LD LD INC INC INC JP PUSH LD ADD POP DEC JP	C.12H B.76H A.(DE) (HL).A HL DE B. NZ-LOPI BC BC, 884AH HL. 8C C. C. NZ-LOP2 (MRAM). A	
B92C	80888888	CHRDAT	ADB	88H.88H.88H.98H	.88H,80H
8938	9889			88H.86H.18H.88H	
	8888 887FFFFE			88H.7FH.8FFH.8F	EH GGH SGH

в93С	9999			
	897FFFFE		DB	08H,7FH,0FFH,0FEH,07H,00H
8942	0700			
B944	88881888		DB	08H.08H.18H.09H.0FH.00H
B948				
	8881FFC8		DB	08H,01H,0FFH,0C0H,1FH.00H
	1F06 00031FE0		DB	88H.83H.1FH.8E8H.3FH.88H
	3FBB		DU	ben esn ira, ecen, orn bea
	888C1FFF		DB	88H,8CH.1FH.8FFH.8FFH.88H
	FF00			
B95C	00301FFF		DB	00H,30H,1FH,0FFH.0FFH.0DH
	FF88 88C81FFF		DB	98H.8C8H.1FH.9FFH.0FEH.88H
B966			DD	sen.ecen.irn.errn.ercn.sen
	BEFFFFE		ESI.	MAH. AFFH. AFFH. AFAH. BAH. BAH
896C	8888			
	00FFFFC0		DB	98H, 8FFH, 8FFH, 8C9H, 88H, 88H
B972			DB	80H.0FFK.0FFH.88K.88H.88H
B974	88FFFF88		DB	DOH. OFFR. OFFR. GER. BUH. BUH
	883FFC88		DB.	88H.3FH.0FCH.80H.88H.08H
B97E				
	80036C88		DB	00H.03H.0CH.00H.00H.00H
B984	9696		-	
B986 B98A	00838C08		DB	00H.83H.0CH.00H.00H.00H
	007FFFE0		DB	86H.7FH.8FFH.SE8H.88H.83H
8998			0.0	
	89899889		DB	89H.98H.98H.98H.96H.96H
8996	9696			
		i		
8998		SPSAVE	DS	82H
B99A		01 01110	DS	100
8906		STACK:		
			-	
B9D6			END	

ヘリコプターの■を回転させる

へリコプターの影を表示するだけでは面白くありません。そこで、ヘリコプターの質を回転させてみることにしましょう。では、どのようにしたら、ヘリコプターの質が回転しているように見えるでしょうか。図4-10 (P.174) のバターンを繰り返せば良いのです。

A. B. C. Dの4つのパターンを定期的に繰り返して CRT 上に表示すれば、 回転しているように見えます。B. Dは同じパターンなので3つのパターンで 済みます。

図4-10 ヘリコプターの■を回転させる



プログラムも簡単です。A、B、C、Dのパターン・データを順番に同じと ころに表示させるだけです。1つのパターンから他のパターンに移るときに時 間待ちしていますが、この待ち時間を調整することによって回転する速さをコ ントロールできます。リスト4-Xにプログラムを示します。

		GRPH3 ^J=7*9- / N# / #47> PROGRAM NAME> Grph3.e				
			***		**********	
B988 E888 428B E626 885C 885C 885E 885E		START: GRAMS: CLS: MON: GRAMB: GRAM1: GRAM2: MRAM:	EQU EQU EQU EQU	88988H 8E868H 428BH 8E626H 5CH 5DH 5EH 5FH		
			ORG	START		
8998	CD8B42		CALL	CLS	:カーソル OFF	
	ED73C1BA 31FFBA		LD LD	(SPSAVE).SP SP.STACK	: ケーンサーイ / スタック タイヒ : アラタニ スタック / セッティ	
898C 898D 898F 8912 8915	8683 78 FE83 C21FB9 2188E8 117DB9 CD5EB9	MAIN: KAITEN:	CP JP LD	B.03H A.B 03H NZ.CD2 HL.GRAMS DE.CDATA1 DISP		

```
8918 85
                          DEC
B91C C349B9
                          JP
B91F 78
                          LD
8928 FE@2
8922 C23289
8925 2188E8
8928 11E989
                                NZ.CD3
                          JP
                                 HL. GRAMS
                                 DE, CDATA2
B92B CD5EB9
B92E 85
B92F C349B9
                          DEC
                          JP.
                                WAIT
B932 2100E0
                 CD3:
                         LD HL.GRAMS
LD DE.CDATA3
CALL DISP
B935 1155BA
B938 CD5EB9
B93B CD4FB9
                          CALL WAT
893E 2188E8
                          1.D
                                HL. GRAMS
                          LD DE.CDATA2
CALL DISP
8941 11E9B9
8944 CD5EB9
                          LD B.03H
8949 CD4FB9
                  WAIT: CALL WAT
894C C38CB9
                          JP KAITEN
B94F D5
                  WAT:
                          PUSH DE
                                                        こうこかい マキ
B950 160F
B952 1EFF
B954 1D
                          I.D
                                D. REH
                  W2:
B955 C254B9
                           JP
                                NZ. W1
8958 15
8959 C25289
895C DI
                                NZ.W2
                           JP.
                           POP
                                DE
B950 C9
895E F3
895F D35E
8961 C5
                  DISP:
                                                         19197 / Early 1-30 7955 NO
                          OUT (GRAW2) A
                                                         : 57- / 40 2" 1->
                           PUSH BC
8962 BE12
8964 B686
                  LOP2:
                                B.86H
8956 1A
                  LOP1:
                                A. (DE)
8967 77
8968 23
8969 13
                                (HL), A
                           1 NO
                                Ht.
                                R
896B C266B9
                          JP
                                 NZ, LOP1
B96E C5
B96F 614A88
                                 BC. BB4AH
B972 09
                                 HL. BC
                                BC
8974 BD
B975 C264B9
                          JP
                                NZ,LOP2
                          POP
                                BC
                          OUT
                               (MRAM) . A
                                                        1 x45 xE9- 5 25
                          13
                                                         170m2 DK
897C C9
                        RET
```

■ 4 音 グラフィックスの世!

aonn		ED LELL DE	2011 2011 2011 2011 2011 2011
B97D B981		CDATA1:DB	89H, 88H, 88H, 88H, 88H
	00001800	-	88H. 88H. 18H. 88H. 88H, 88H
B987		Page 1	ebii.ebii.ibii.ebii.ebii.beii
	ØBFFFFFF	DB	88H,8FFH,8FFH,8FFH,88H,88H
B98D	8888		
	00FFFFFF	DB	88H.0FFH.8FFH.8FFK.87H.88H
B993		_	
		1000	09H.00H.18H.00H.0FH.00H
8999	0001FFC0	0.0	89H.81H.8FFH.8C8H.1FH.08H
B99F		DD	enninin.errn.econ.irn.een
	00031FE0	DB	88H.03H.1FH.0E6H.3FH.80H
B9A5			
B9A7	000C1FFF	DB	88H.8CH.1FH.8FFH.9FFH.08H
	FF06		
	8838)FFF	DB	00H.30H.1FR.0FFH.0FFH.08H
B9B!	88C81FFF	D.D.	88H. 8C8H. 1FH. 8FFH. 8FEH. 88H
	FE00	DB	DEN.DCON.IPS.BFFH.OFEN.OBA
	DØFFFFE	na na	88H. 0FFH. 0FFH. 0E8H. 88H. 80H
BBBD			
B9BF	u0FFFFC0	DB	98H. 9FFH. 0FFH. 0C0H. 09H. 08H
B9C3			
	00FFFF80	DB	00H.0FFH.0FFH.80H.00H.00H
B9C9		20	88H.3FH.8FCH.88H.00H.80H
BBCF	003FFC00	UD	BEH. JEH. BECH. BEH. OUR. BEH
	80038C88	DB	00H.03H.0CH.00H.00H.00H
	0000		
	09C30C00	DB	08H.0C3H.0CH.00H.09H.09H
	9888		
	097FFFE0	DB	08H.7FH.0FFH.0E0H.00H.00H
	89999988	p.n.	88H.89H.88H.98H.98H.98H
B9£3	9696	DB	BER-EDH-BEH-BEH-BEH
Dar	0000	:	
		CDATA2:DB	88H.80H.89H.88H,68H,98H
B9ED	9686		
B9F3	00001900	08	88H.88H.18H.88H.88H.88H
	8080FF80	DB	98H.90H.9FFH.88H.98H.88H
B9F9		00	0011.0011.011111.0011.0011.0011
	BBBBFFBB	DB	00H.00H.0FFH.00H.07H.00H
B9FF	0700		
	00001300	DB	98H.98H.18H.86H.8FH.89H
	0F00		
BA07 BA0B	8001FFC0	DB	88H.81H.8FFH.8C8K.1FH.88H
	80031FE0	D/B	00H.03H.1FH.0E8H.3FH.00H
	3F00	νD	Man's Add the Beau and Box
	888C1FFF	DB	00H.0CH.1FH.0FFH.0FFH.08H
BAI7	FF00		
	00301FFF	DB	88H,38H,1FH,8FFH.8FFH.88H
	FF00		
	00C01FFF	08	00H.0C0H.1FH.0FFH.0FEH.90H
	FE00	nn.	60H.8FFH.0FFH.6E0H.88H.08H
0.1423	DULLEFEE	DU	Composition of the Control of the Co

```
8A29 8888
BA2B BOFFFFCB
                   DB 88H.8FFH.8FFH.9C8H.88H.88H
BA2F 6666
BA31 BREEFERR
                   DB 88H.8FFH.8FFH.88H.88H.88H.88H
BA35 8686
BA37 003FFC00
                   DB 88H.3FH.8FCH.88H.88H.88H
BA3B 8666
BASD 88838C88
                   DB 88H,83H,8CH,88H,89H,88H
BA41 8886
BA43 00830C00
                   D8
                        00H.83H.0CH.00H.09H.00H
BA47 8898
BA49 887FFFF8
                   DB 00H.7FH.0FFH.0E6H.00H.00H
BA4D 0000
BA4F 96998988
                   DB 88H, 68H, 88H, 88H, 88H, 86H, 86H
BA53 8888
BA55 80808000 CDATA3: DB 80H. 88H. 88H. 88H. 88H. 88H
BA59 0000
BA5B #### 188#
                   DB 40H.28H.18H.00H.08H.08H
                        88H. 68H. 18H. 88H. 86H. 08H
BASI BARRISHS
                   OR.
BA65 8888
BAS7 80801868
                   DB 80H,80H,18H,80H,07H,00H
BA68 8788
BA6D 00001800
                   MAN ANN ISH REH REH ANN
BA71 0F00
BA73 0001FFC0
                   ## 98H.81H.8FFH.8C8H.1FH,88H
BA77 1F68
BA79 88831FF8
                   DB 09H.83H.1FH.0E0H.3FH.09H
BA7D 3F88
                   DB 00H.0CH.1FH.0FFH.0FFH.00H
BAZE BERCIFFE
BARS FERR
                   DB 08H.36H, LFH. 0FFH, 0FFH.09H
BARS BRIGHTEF
BASS FFEE
BASS 68C61FFF
                   DB 86H, 8C8H, LFH, 8FFH, 8FEH, 86H
BARE FERR
BA91 00FFFFE0
                   DB 88H.8FFH.8FFH.8E6H.80H.88H
BA95 0000
                   DR 08H.8FFH.0FFH.0C8H.88H.88H
BAS7 80FFFFC8
BASE SEES
                        86H. 8FFH. 8FFH. 88H. 80H, 88H
BASD BREFFERR
                   DB
BAAL BRRR
                        89H, 3FH, 0FCH, 88H, 80H, 80H
BAA3 863FFC88
                   DB
BAA7 8888
                   DB 68H.03H.8CH.00H.06H.08H
BAA9 88838C88
RAAD ARRA
                   OR 88H.83H.8CH.88H.98H.98H
BAAF 00830C00
BAB3 8898
                    DB 06H.7FH.8FFH.0E6H.80H.86H
BARS MRZEFEEN
BARS GREE
                   DB 80H.80H.88K.88H.80H.80H
BARR 80890888
BABE MAN
BAC1
              SPSAVE: DS
                     DS 50
BAC3
BAFF
              STACK:
BAFF
                   END
```

ヘリコプターを移動させる

ヘリコブターをキーで移動させるプログラムを作ってみましょう。まずは簡 里に4万向(上,下,左,石)に移動させてみましょう(図4-11).

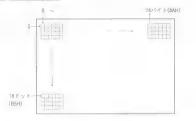
図4-11 4つの方向



■難しそうだけど実は簡単

難しそうにみえますが、楽にできます。というのは、以南等3 常3 節でキー 人力のプログラムを作っているからです。このプログラムを一部予証しして、 ヘリコブター移動のプログラムを作ります。以前のプログラムでは、移動キャ フクタが1 バイト・データでしたが、今回のは108バイトのキャラクタになる点 だけが進います。1 バイト・データを表示するのも、108 バイトのデータを表示 するのも、アルゴリズム自体には変りありません。1 バイト・データを表示す

図4-12 ヘリコプターの移動幅:



る代りに、108パイトを表示するサブルーチンを呼ぶだけのことです。 ヘリコプターの移動転は、

■の移動幅 8Dパイトーヘリコプターの横のパイト圏 (6パイト)

=74/57 F (4AH)

綴の移動幅 200ドットーへリコプターの縦のドット■ (18ドット)

= 182ドット (B6H) ······プログラム上では B5H

となります。ヘリコプターの回転線はいつも回転していなければならないので、 キー入力の部分はサブルーチンとなります。

■ヘリコプター ■ 4 方向に移動させ■

では、ヘリコブターを4方向に移動させるプログラムをリスト4-4 に示しま す。このプロプラムでは、上下方向はドット単位で移動しているのでスムーズ に移動しますが、横方向は1バイト (8ビット)単位の移動なので、動きが維 になっています(図4-13)



なお、横方向をスムーズに移動させるプログラムは第5章で説明しますので、 そちらを御覧ください。

	GRPH4 1	107* ター ノ バネ ノ	カイテン ト キーニュウリョク	
	:PROGRAM NA	ME> Grph	. е	
			*****	******
	:			
8908	START: EQU	8B989H		
C669	GRAMS: EQU	9C998H		
6F6B	COLOR: EQU	6F6BH		
4288	CLS: EQU	428BH	:カーソミ ケス	
E626	MON: EQU	8E626H		
0.05C	CRIMA: FOU	SCH		

GRAM1: EDU 5DH

865D

第4章 グラフィックスの世界

005E 005F		GRAM2: MRAM:	EQU	5EH 5FH	
			ORG	START	
	326EBB 31ACBB		LD LD	(SPSAVE).A SP.STACK	: ケニンサニイ ノ スタック ? : アラタニ スタック ヲ セッ?
B906 8909 B908 B908 B910 B913 B915 B918 B918	011950 3E01 32B2E6 3E19 32B3E6 3E68 3E68 32B4E6 32B8E6			BC.5819H A.01H (0E682H),A A.19H (0E683H),A A.0E8H (0E664H),A A.00H (0E688H),A A.00FFH (0E689H),A	
	CD6B6F		CALL	COLOR	
B925	CD8B42		CALL	CLS	∥h-V⊩ OFF
B92B	2188C8 118888 3E27	MAIN:	LD LD	HL.GRAMS DE.0000H A.27H	:ハシンメノ ヒョウシンイチ
8938 8931 8932 8933 8936	23 14	ML1:	JP LD	HL D A NZ,ML1 A.108 BC,8950H	
893B 893C 893D	89 1C	ML2:	ADD INC DEC	HE.BC E A NZ.ML2	
B941	8603	,	LD	В.03Н	
B944 B946 B949 B94A	78 FE03 C257B9 E5 D5 112ABA	KAITEN	CP JP PUSH PUSH LD	DE . CDATAL	
B951 B952 B953			POP	DISP B DE HL WAIT	
895A 895D 895E 895F	FE82 C26BB9 E5 D5 1196BA CD0BBA	CD2:	LD CP JP PUSH PUSH LD CALL DEC POP	B2H NZ.CD3 HL DE DE.CDATA2 DISP B	

```
HE.
R967 E1
                      JP
                           WAIT
8968 C384B9
                      PUSH HL
896B ES
896C D5
                      PUSH DE
B960 1162BB
                      LD DE.CDATA3
                      CALL DISP
                      CALL WAT
8973 CDFCB9
8976 D1
                      POP DE
                      POP HI.
                      PUSH HL
B978 E5
                      PUSH DE
B979 05
                      LD DE, CDATA2
897A 1196BA
                      CALL DISP
B97D CD98BA
                      LD B. B3H
B980 8683
                      POP DE
B982 D1
                      POP HL
8983 E1
8984 CDFCB9
               WAIT:
                      CALL VAT
B987 C398B9
                      JP
                           KIY
               WKIY:
                      IN
                           A.(09H)
B98A D809
                      AND BIH
B98C E681
B98E C243B9
                      JP
                           NZ.KAITEN
                           SP. (SPSAVE)
                                              : F1 29v7 = F1"2
B991 ED786E88
                      1.D
                      JP MON
B995 C326E6
                      PUSH BC
8998 C5
                      IN A. (00H)
B999 D866
                           B.A
8998 47
B99C E684
B99E CAB8B9
                      AND 84H
                      JP
                           Z, DOWN
                      LD A.B
 B9A1 78
89A2 E619
89A4 CAC8B9
                      AND LOR
                      JP Z.LEFT
 B9A7 7B
                      LD A.B
                      AND 48H
 B9A8 E649
 BSAA CADSBS
                      JP
                            Z.RIGHT
 BSAD DB61
                       IN
                            A. (81H)
                      AND
                           BIH
 89AF E601
 8981 CAFARS
                      JP
               KIYPOP: POP
                            80
8984 C1
B9B5 C38AB9
                            WKIY
 8988 7B
               DOWN:
                            A.E
                      CP
 8989 FEB5
                            0B5H
                      JP
                            Z.KIYPOP
 89BB CAB4B9
 B98E C5
B98F 015000
                      PUSH BC
                            BC,0050H
                      ADD
                            HL . BC
 B9C2 89
                      202
                           BC
 89C3 C1
                      INC
```

第4章 グラフィックスの世界

```
JP KIYPOP
B9C5 C3B4B9
 B9C8 7A
              LEFT: LD A.D
              CP 88H
 B9C9 FE00
 89CB CAB4B9
                     JP Z.KIYPOP
DEC HL
 BSCE 28
B9CF 15
                     DEC D
PUSH BC
LD BC.8858H
AND A
 89F1 815888
 B9F4 A7
 89F5 ED42
                     SBC HL.BC
POP BC
 B9F7 C1
B9F8 1D
                  DEC E
JP KIYPOP
 B9F9 C384B9
B9FC D5 WAT: PUSH DE
B9FD 1607 LD D.07H
B9FF 1EEF W2: LD E.0EFH
BA01 1D W1: DEC E
                                         :シ*カン マチ
 BA02 C201BA
BA05 15
BA06 C2FFB9
BA09 D1
                     JP NZ.W1
                      DEC D
                     JP NZ.W2
                     POP DÉ
                     RET
BARA C9
                     DI : 2*ラフ / ヒョウシ ルーチン フリコミ NO
OUT (GRAM2).A : カラー ノ イロ ク*リーン
 BA14 77
                     LD (HL), A
 BA15 23
                     INC HL
BA16 13
BA17 85
BA18 C213BA
                     INC DE
                 INC DE DEC B JP NZ.LOP1 PUSH BC LD BC.0044AH ADD HL.BC PDP BC C JP NZ.LOP2 POP BC
BAIB C5
BAIC 814A88
BAIF 89
BA21 0D
BA22 C211BA
BA25 C1
BA25 C1
```

§2 グラフィックスのプログラムを作る

BA25 D35F 0UT (MRAM).A : X/C x8y = 2 %	
BAZA 6888686 CDATAI:DB 68H.88H.96H.88H.88H.88H.88H.88H.88H.88H.88H.88H.8	
BAZE 8888 BAZE 8880 ABB DB 88H.88H.18K.88H.86K.86H.86H BAZE 8880 ABB DB 88H.8FFH.8FFH.8FFH.8FFH.8FFH.8FFH.8FFH.	
BAZE 8080 BA39 8080 808 DB 88H.88H.18K.86H.86H.86H.88H BA34 808FFFFF DB 88M.8FFH.8FFH.8FFH.8FFH.88H BA3C 80FFFFFF DB 88H.8FFH.8FFH.8FFH.8FH.8FH.8FH.8FH.8FH.8	
BA39 8880 888	
BA36 08FFFFF DB 08H.8FFH.0FFH.08H.08H BA30 08FFFFF DB 08H.8FFH.0FFH.0FH.08H BA30 07000 DB 08H.8FFH.0FFH.0FH.08H	
BA36 8880 BA3C 80FFFFFF DB 88H.8FFH.8FFH.8FFH.87H.88H BA48 8788	
BA40 8700	
BA42 80801800 DB 60H.00H.18H.00H.0FH.00H BA46 0F00	
BA48 8001FFC8 DB 08H.01H.0FFH.0C8H.1FH.00H BA4C 1F08	
BA4E 00031FE0 DB 00H.03H.1FH.0E0H.3FK.00H	
BA52 3F08 BA54 000C1FFF DB 80H.0CH.1FH.8FFH.0FFH.00H	
BA58 FF00 BA5A 00301FFF DB 00H,30H,1FH.0FFH.0FFH.00H	
BASE FF00 BASE 00C01FFF DB 00H.0C0H.1FH.8FFH.0FEH.00H	
BA64 FE00 BA66 00FFFFE0 DB 80H.0FFH.0FFH.0E0H.00H	
BAGA 0000	
BASC 00FFFFC0 DB 00H.0FFH.0FFH.0C0H.00H.00H	
BA72 00FFFF80 DB 00H.0FFH.0FFH.00H.00H	
BA78 883FFC00 DB 80H.3FH.8FCH.88H.88H.60H BA7C 0880	
BA7E 00038C80 DB 80H.83H.0CH.80H.00H	
BA84 B0C38C89 DB B0H. 0C3H. BCH. 00H. 00H, 88H	
BASA 807FFFE0 DB 89H.7FH.8FFH.8E8H.88H	
BASE 8080 DB 88H.88H.88H.88H.88H.88H	
BA94 8880	
BA95 88888888 CDATA2:DB 88H.88H.88H.88H.88H.88H	
BA9A 0000 BA9C 00001800 DB 00H.00H.18H.00H.00H	
BAAG 0000 BAAG 0000FF00 DB 00H.08H.0FFH.00H.00H.00H	
BAA6 0000 BAA6 0000FF00 DB 00H.00H.0FFH.00H.07H.00H	
BAAC 8788	
BAAE 08081880 DB 00H.00H.18H.00H.06H BAB2 0F00	
BAB4 6861FFC0 DB 86H.81H.8FFH.8C8H.1FH.8CH BAB8 1F69	
BABA 08031FE0 DB 00H,03H,1FH.0E0H,3FH,00H BABE 3F00	
BACS SESCIFFF DB SSH.SCH.IFH.SFFH.SFFH.SOH	

第4章 グラフィックスの世界

BAC4	FF86		
	00301FFF	DB	86H.30H.1FH.8FFH,8FFH,80H
	FF88		
	9BC91FFF FE88	100	00H.0C0H.1FH.0FFH.0FEH.00H
	00FFFFE0	D.O.	ARIL APPLI APPLI APAIL ARIL ARIL
	8808	no	08H.0FFH.0FFH.0E0H.80H.80H
	00FFFFC0	DB	88H, 8FFH, 8FFH, 8C8H, 88H, 88H
	8868	55	DUNION CONTROL CONTROLS
BADE	00FFFF80	DB	88H.8FFH.0FFH.80H.88H.88H
	8908		
	003FFC88	DB	00H.3FH.0FCH.00H.00H.00H
	0969		
	00030000	DB	88H.83H.8CH.88H.88H.88H
	8868	1000	80H.8C3H.8CH.88H.88H.88H
	8868	1000	EGH. DC3H. SCH. GGH. GGH. GGH
	007FFFE0	DB	86H.7FH.8FFH.8E8H.88H.88H
	0000	00	001111111111111111111111111111111111111
	00000000	DB	89H.89H.89H.99H.98H.98H
8800	8868		
		;	
		CDATA3:DB	88H.80H.80H.88H.68H.68H
	8860 68961806	100	2011 2011 1011 2011 2011 2011
	8988	100	88H.88H.18H.88H.89H.89H
	63081800	DB	88H.88H.18H.88H.88H.89H
	8888	00	55111551111511115111151111
	88818888	DB	88H.88H.18H.88H.87H.86H
	9789		
	88818988	DB	68H,88H.18H.86H.8FH.86H
	8F89		
	1F00	DB	00H.01H.0FFH.0C0H.1FH.00H
	88831FE8	D.D.	80H.83H.1FH.8E8H.3FH.80H
	3F80	DD	OBH. GON. IFH. BEEN, OFH. BEH
	000CIFFF	DB	88H. 9CH. 1FH. 0FFH, 0FFH, 08H
BB30	FF86		
	00301FFF	DB	86H.36H.1FH.6FFH.8FFH.89H
	FF00		
	80C01FFF	DB	88H.BCBH.IFH.BFFH.BFEH.88H
	FE00 00FFFFE0	P.O.	88H. 9FFH. 8FFH. 8E8H. 88H. 98H
		UD	DDR. BFFR. DFFH. BECH. BOH. BBH
	889FFFFC8	200	ASH APPLI APPLI ACALL BALL BALL
	8869	np	00H.0FFH.0FFH.0C0H.00H.00H
	00FFFF80	DB	00H.0FFH.0FFH.88H.00H.88H
	8868	00	OBJECT II. BETTH. BUIL DOIL DOIL
	883FFC68	DB	00H.3FH.0FCH.08H.00H.08H
	8969		
	88638088	DB	08H.83H.8CH.80H.80H.88H
	8989		
	88038088	DB	00H.8C3H.0CH.00H.00H.00H
	8889 887FFFE8	DB	00H.7FH.0FFH.8E0H.88H,08H
B866		<i>DD</i>	COMMITTER CONTRACTOR C
	888888888	DB	88H.88H.88H.88H.88H.88H
BBSC			

BBGE SPSAVE: DS 02H BB70 DS 60 BBAC STACK:

BBAC END

■ヘリコプターを8方向に移動させる

また、8 方面に移動させるプログラムもリスト4-5に示します。

これで第4章を終りますが、グラフィックスの原理や表示のやり方が理解で きたでしょうか、次章は横方向もドット単位で移動させますので、動きがもっ とスムーズになります。

●リスト4-5 ヘリコプターの画画(8方向)■

			****	*****	****	*****	***
		: GRPH-	-5 51	17°9- / 112 /	カイテント	キャニュウリョ	2 88.200
		1					
		: PROGR	AG NA	WE> Grph!	5.e		

			****	****	****	******	*******
B900 C000 6F6B		START: GRAMS: COLOR:	EQU	0B900H 0C800H 6F6BH			
4288 E626 865C 885D		CLS: MON: GRAMB: GRAM1:		428BH 9E626H 5CH 5DH		:カーソル ラ	Z.
805E 885F		GRAM2: MRAM:		5EH 5FH			
			ORG	START			
	ED73E8BB 3126BC		LD LD	(SPSAVE).SP SP.STACK		: 25m2 /	タイヒ
B90A B90C B90F B911 B914	811958 3E81 32B2E6 3E19 32B3E6 3EE8 32B4E6			BC.5019H A.01H (0E6B2H1.A A.19H (0E6B3H).A A.0E8H (0E6B4H).A			
8919 891B	3E00 32B8E6		LD	A.80H (0E6B8H).A			
	3EFF		LD LD	A.OFFH			
	32B9E6 CD6B6F			(ØE6B9H).A COLOR			
3000	CDODGI	:	GHUU	COLOR			
B926	CD8B42		CALL	CLS			
Ragg	2189C8	MAIN:	LD	HL. GRAMS			
	110000		LD	DE.0000H		: h5" XJ 1	ヒョウシェイチ

第4章 グラフィックスの世界

```
B92F 3E27
                      LD A.27H
INC HL
                       INC
              MLI:
B931 23
                        INC
                             65
8933 3D
                        DEC
                             Α
                       JP
                             NZ.ML1
B934 C231B9
8937 3E64
                             A.108
B939 815000
                             BC,0050H
B93C 89
               ML.2:
                       ADD
                             HL.BC
                        INC
                             E
                             A
893F C23CB9
                        JP
                             NZ., ML2
                        LD B. 03H
8942 0693
               KAITEN:LD A.8
CP 83H
JP NZ.CD2
B944 78
B945 FE03
B947 C258B9
B94A E5
B94B D5
B94B D5
B94C 11A4BA
B94F CD85BA
               PUSH BE
PUSH DE
LD DE.CDATAI
CALL DISP
                       DEC B
POP DE
POP HL
JP WAIT
B954 E1
B958 78
B959 FE02
                     LD A.B
CP 02H
JP NZ.CD3
                CD2:
 B95B C26CB9
B95E E5
B95F D5
B960 11108B
                        PUSH HL
                       PUSH DE
LD DE.CDATA2
CALL DISP
 B963 CD85BA
 8966 85
                       DEC B
POP DE
POP HL
 8967 EI
B968 E1
 B969 C385B9
                       JP WAIT
PUSH HL
                       PUSH DE
 897A D5
597B 111@BB
                       LD DE.CDATA2
                       CALL DISP
LD B.03H
POP DE
B97E CD85BA
B981 0603
 B983 D1
                        POP
                            HL
                WAIT: CALL WAT
 B988 C399B9
                     JP KIV
                WKIY: 1N A. (89H)
B98B DB89
                AND BIH
B98D E681
 B98F C244B9
                       JP
```

§2 グラフィックスのプログラムを作る

0002	E078E888		LD	SP. (SPSAVE	()	
0000						
8996	C326E6		JP	MON		
B999	rs.	KIY:	PUSH	BC		
	D868		LN	A. (88H)		1.1
B99C			L.D	B.A		
	E683		AND	92H		
	CAD289		JP	Z.LDOWN		
past	CHUZUO					
B9A2	78		LD	A-B		1.2
	E684		AND	0.4H		
	CAE9B9		JP	Z.DOWN		
						: 3
B9A8			LD	A.B		1.3
	E688		AND	88H		
B9A6	CAF9B9		JP	Z.RDOWN		
		1				
20.0	26	1	1.0	4 2		
BBAE				A.B		:4
B9AF	CALBBA		AND	16H		
0301	CATOBA		JP	Z.LEFT		
8984	7.8		LD	A.B		: 6
	E646		AND	49H		. 0
	CA21BA		JP	Z.RIGHT		
B9BA				A.B		17
	E688		AND	80H		
898D	CA32BA		JP	Z.LUP		
		i				
B9C0	DBGI		LN.	A.(81H)		: 8
Bacc			LD	B.A		
B9C3			AND	011		
	CA4BBA		JP	Z.UP		
B9C8			LD	A.B		:9
	E602		AND	82H		
B9CB	CA5D8A		JP	Z.RUP		
noor	C1	;		0.0		
89CE	C1	KIALOL:	FUP	BC		
89CF	C38B89		JP	WKIY		
0.001						
B902		LOOWN:		A.E		
	FEB5		CP	6B5H		
	CACEB9		JP	Z.KIYPOP		
B9D8			LÐ	A,D		
	FE98		CP	86H		
	CACEB9		JP	Z,KIYPOP		
B9D€	C5		PUSH			
	814F88		LD	BC.004FH		
B9E2			ADD	HL.BC		
			POP	BC		
B9E4				E		
B9E3 B9E4 B9E5			DEC	D KIYPOP		

第4章 グラフィックスの世

89E9 78 89EA FEB5 89EC CACEB9 89EF C5 89FB 815888 89F3 89 89F4 C1 89F5 IC 89F6 C3CEB9	DO₩N:	LD CP JP PUSH LD ADD POP INC JP	A.E 0B5H Z.KIYPOP BC BC.0050H HL.BC BC E KIYPOP
89F9 7B 89FA FEB5 89FC CACEBS 89FF 7A 8A80 FE4A 8A82 CACEBS 8A85 C5 8A86 815180 8A89 89 8A84 C1 8A8B 1C 8A8B 1C 8A8B 1C 8A8C 14 8A8D C3CEBS		LD CP JP LD CP JP PUSH LD ADD POP INC INC	A,E 8B5H Z,KIYPOP A,D 4AH Z,KIYPOP BC BC,0851H HL.BC BC E D KIYPOP
BA1B 7A BA11 FEBB BA13 CACEB9 BA16 2B BA17 15 BA18 CD76BA BA1B CD76BA BA1E C3CEB9	LEFT:	LD CP JP DEC DEC CALL CALL JP	A.D 88H Z.KIYPOP HL WAT WAT KIYPOP
BA21 7A BA22 FE4A BA24 CACEB9 BA27 23 BA28 14 BA29 CD76BA BA2C CD76BA BA2F C3CEB9	RIGHT:	CP JP INC INC CALL	A.D 4AH Z.KIYPOP HL D WAT WAT KIYPOP
BA32 7B BA33 FE00 BA35 CACEB9 BA36 7A BA39 FE00 BA38 CACEB9 BA3E C5 BA47 815100 BA42 A7 BA43 ED42 BA45 C1 BA46 1D BA47 15 BA48 C3CEB9	LUP:	LD CP JP LD CP PUSH LD AND SBC POP DEC JP	A.E 00H Z.KIYPOP A.D BBH Z.KIYPOP BC BC,51H A HL.BC BC E D XIYPOP
BA4B 7B BA4C FE00 BA4E CACEBS BA51 C5	ÜP:	LD CP JP PUSH	A.E 88H Z.KIYPOP BC

§2 グラフィックスのプログラムを作る

```
BAS2 015000
                    LD BC.8859H
BASS A7
                     AND A
BA56 ED42
                     SBC HL., BC
BA58 CI
                     POP
                         RC
                     DEC E
BA59 1D
BASA C3CEB9
                     JP
                          KIYPOP
BASD 7B
              RITE:
                    1 D
                     LD A.E
CP B6H
BASE FERR
                     JP
BAGD CACEB9
BA63 7A
                         A.D
BAG4 FE4A
                          401
BASS CACEBS
                     JP
                         Z.KIYPOP
BA69 C5
                    PUSH BC
BAGA 814F08
                    LD BC.884FH
BAGE ED42
                    AND A
SBC HL.BC
                     POP BC
BA70 CI
BA71 1D
                     DEC
                          E
BA72 14
                     JP KIY
BA73 C3CEB9
                         KIYPOP
              WAT: PUSH DE
                                             :シーカン マチ
BA76 D5
                     LD D.04H
LD E.0EFH
BA77 1504
              ₩2:
₩1:
BA79 1EEF
                          E. ØEFH
                     DEC E
BA7B ID
BA7C C27BBA
BA7F 15
                      JP
                          NZ.WI
                      DEC
                         D
BA80 C279BA
                      JP.
                          N7. W2
                     POP DE
BA83 D1
BA84 C9
                     RET
                                            : 5° 57 / Eagon &-Fo 7932 NO
              DISP:
BASS F3
                     DI
                     OUT (GRAM2), A
                                             : #5- / (0.019-5
BA86 D35E
BA88 C5
                     PUSH BC
BA89 8E12
                     LD C.12H
                         B. 86H
BA8B 8606
              LOP2:
                     LD
BASD 1A
              LOP1:
                          A. (DE)
BABE 77
BABF 23
                          (HL) A
                     INC
                          HI
BA99 13
                     INC
                          DE
BA91 05
                          B
                          NZ.LOPI
BA92 C28DBA
BA95 C5
                     JP
                     PUSH BC
BA96 814A88
                     I.D
                           BC.804AH
BASS III
                     ADD
                           HL. BC
                         BC
                     POP
BASA CI
BA9B
                    DEC
BA9C C28BBA
                    JP NZ
POP BC
                          NZ.LOP2
BASE C1
                     OUT (MRAM), A
                                             : メイン メモリー ニ スN
BAA# D35F
                                              : 9115 PK
BAA2 FB
                     EI
BAA3 C9
```

BAA4 86808088 CDATA1:DB 86H.86H.86H.86H.86H.88H

第4章 グラフィックスの世界

BAAA	00001800	D	п	00H.00H.18H.00H.00H.00H
	6666			2211,0011,1011,0211,0211,0211
	00FFFFFF 0000	D	В	08H.0FFH.0FFH.0FFH.06H.00H
BAB6	08FFFFFF		101	88H.0FFH.8FFH.8FFH.87H.08H
	00001800	D	В	88H.88H.18H.88H.8FH.88H
BAC6 BAC2	0F00 0001FFC0	D	В	00H.01H,0FFH.0CDH.1FH.00H
BAC6	1F00			
BACC	00031FE0		В	08H,03H,1FH,0E0H.3FH,00H
	888CIFFF FF88	D	В	88H.0CH.1FH.8FFH.8FFH.88H
BAD4	00301FFF FF00	Ð	В	00H.36H.1FH.0FFH.0FFH.00H
BADA	00C01FFF	D	В	80H.0C0H.)FH.0FFH.0FEH.00H
	PERS REFFFER	0	В	88H.8FFH.8FFH.8E8H.88H.88H
BAE4	8888 88FFFFCB			
BAEA	9886			08H.8FFH.0FFH.8C8H.80H.0HH
	88FFFF88	D	В	00H.0FFH.0FFH.86H.00H.00H
BAF2	883FFC88	D	В	00H.3FH.0FCH.00H.00H.8UH
BAF8	80030C88	D	В	08H.83H.8CH.80H.88H.08H
BAFC	9996 98C39C89	D	В	00H.0C3H.0CH.00H.00H.00H
BBØ2	0000 007FFFE0			08H.7FH.0FFH.0E8H.08H.08H
8888	8888			
	88888888	U	В	88H.88H.86H.86H.89H.88H
	88888888	CDATA2:D	В	00H.00H.00H.00H.00H.00H
BB16	00001800	D	В	88H.88H.18H.80H.80H.80H
BBIC	0000 0000FF00	D	В	88H.88H.8FFH.88H.88H.88H
	8888 8888FF88	D	В	82H.88H.8FFH.88H.07H.88H
	0700		D	68H.88H.18H.86H.8FH.66H
BB2C	8F88			
	0001FFC0	D	В	08H.01H.0FFH.0C6H.1FH.08H
	00031FE0	D	В	00H.03H.1FH.0E0H.3FH.00H
BB3A	000CIFFF	D	В	88H, BCH, LFH, 0FFH, 0FFH, 08H
BB48	FF08 06301FFF	D	В	80H.39H.1FH.0FFH.0FFH.60H
	FF08 08C01FFF	D	В	80H.0C8H.1FH.0FFH.0FEH.00H
BB4A	FE00 00FFFFE0			00K.0FFH.0FFH.0E8H.00H.00H
BB50	8888			
BB52	DEFFFFCE	D	R	00H.8FFH.0FFH.8C8H.00H.00H

第4章 グラフィックスの世界

```
SR56 OBBB
BB58 MMEEEEB8
                  DB
                         ANH AFEH AFEH ANH ANH ANH
8B5C 0000
BBSE 003FFC00
                   DB
                         88H.3FH.8FCH.88H.98H.88H
BB62 6686
BBS4 ROBSECSE
                    1000
                         MAH. MSH. MCH. MAH. MAH. AMH
BB68 0000
BB6A 80C38C00
                   DB GGH. 8C3H. DCH. 66H. 66H. 66H
BBSE 0000
BB70 897FFFF8
                  DB 00H.7FH.0FFH.0EBH.09H.90H
BB76 80898888
                   DB 89H, 80H, 68H, 98H, 66H, 66H
BB7A 0888
BB7C 88888888 CDATA3:DB 89H.88H.88H.88H.88H.88H
BB82 00001800
                  DB 86H.86H.18H.68H.80H.86H
8886 9999
BB88 RRRR1888
                 DB GRH. 89H. 18H. 99H. 99H. 69H
B88C 0000 800
                   00H, 80H, 18H, 80H, 97H, 80H
8892 0700
8894 10901800
                  DB 88H.08H.18H.80H.0FH.06H
BB98 0F00
BB9A 8001FFC0
                  DB 88H.81H.8FFH.8C8H.1FH.88H
BB9E IF00
                  OB 86H.83H.1FH.6E6H.3FH.66H
88A6 989CIFFF
                  08
                        88H. SCH. LFH. 8FFH. 8FFH. 88H
BBAC 98381FFF
                  D8 00H,30H,1FH,0FFH,0FFH,08H
BBB9 FF66
8882 GOCDLEFF
                  DB 88H.8C9H.1FH.8FFH.8FEH.88H
                  DB 88H.8EFH.8EFH.8E8H.88H.88H
888C 8666
BBBE @@FFFFC@
                   DB 88H.8FFH.8FFH.8C8H.88H.88H
BBC2 0000
BBC4 00FFFF80
                  DB 88H.8FFH.8FFH.88H.08H.89H
88C8 8000
BBCA 983FFC98
                  DB 08H,3FH,6FCH,88H,88H,68H
BBCE 0000
BBD0 80938C08
                  D8 88H.83H.8CH.88H.88H.88H
BBD4 8888
BBD6 88C38C88
                  DB 88H.8C3H.8CH.80H.80H.89H
BBDA 8868
BBDC #87FFFE8
                  DB 88H.7FH.8FFH.8E8H.88H.88H.88H
BBE8 8000
ввез наванава
                   DB 66H,86H,66H,86H,86H,86H
BRES 8888
             SPSAVE: DS 02H
                    DS 68
BC26
BC26
                 END
```



5 スムーズ・スクロール

方向 ドット単位であかす

この章では、先に使ったヘリコブター をさらにスムーズに移動させる方法を考 えます、つまり、横方向もドット単位で 移動させようというわけです。

しかし、VRAMの関係上、8ビットを1 単位として競りなければならないので、 横の動きは極ほど簡単にはいきません。 まず、横方向の移動に必要なビットのシ フト命令から学ん。スムーズな動きの ブログラムを作ります。

また、チキスト画面とグラフィックス 画面を重ね合わせたり、勧きを遠くする ためにDMAを止めたりといったテクニッ クも紹介しています。

スムーズな動きの原理

右ヘシフトするには問題がある

へリコブターをスムーズに動かすためには、縦方向だけでなく機方向もドット単位で動くようにしなければなりません。そこで、ヘリコブターのパターン・データを表示する前は、左右ヘビット単位でシフト (初送り)させます。シフトは簡単なようで結構予問がかかります。まず、右へのシフトを考えてみましょう (図5-1)。

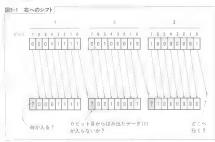


図5-1からわかるように、右ヘシフトする場合には2つ問題があります。

- 1 7ビット目には何が入るか?
- 2. 0ビット目からはみ出したデータは消えてしまうのか?

この2つの問題を解決しなければなりません。そのためには、右ヘシフトする

第5章 スムーズ・スクロール

ときにデータが次のように動いてくれれば良いわけです。

- 1、1バイト目のデータを右へ1ビットシフトしたときには、7ビット目に 0が入る。
- 1パイト目のデータを右へ1ビットシフトしたときには、0ビット目の データが2パイト目のデータの7ビット目に入る。
- 2パイト目のデータの場合は、7ビット目には前のデータ (1パイト目 0ビット目のデータ) が入り、0ビット目のデータは3パイト目のデータの7ビット員に入る。

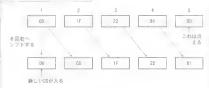
4 3バイト目のデータも、2パイト目のデータと同じシフトを行う。

しかし、この4つ全部を満たしたとしても、まだ問題があります。8回右へ シフトしたら、データは1パイト目と3パイト目のデータが00日になってしま います。限りました。ここで、ヘリコプターのパターンを表示したときのこと を思い出してみましょう。

へリコブターのデータは1列からバイトのデータの集りでした。そのうちの
1パイト目とらパイト目は、すべて00日にしてありました。この00日は、8ビ
ット左右へ移動したときにデータを保存するという役目を持っていたのです。
ですから、左右に1パイトずつ00日のデータをつけておけば、8回までビット
をシフトすることができます(図5-2)、8回を越えた場合はVRAMの番地を1 番地分泌めればよいので、8ビット分のシフトさえできれば大丈夫です。また、 をへのシフトも右へのシフトと同じ原理なのでわかると思います。

■はみ出たデータはどこへ行く

0ビット日のデータを右へシフトすると、はみ出たデータはどうなるのでし 図5-2 左右にバイトずつ00Hをつける



ょうか、このはみ出たデータを保存するフラグがあります。それが Cy(キャリー) です。Cv フラグはF (フラグ) レジスタのひとつです。

■仮想CY(キャリー)レジスタとCy(キャリー)フラグの考え方

CPUの内部にあるF(フラグ)レジスタのフラグは全部で6個ありました。 本来、フラグはブランナ (Branch: 分岐) 機能のためにあり、条件判断に多く 使用きれています。ここでは、フラグの状態が1か0かによって条件を判断し ているだけで、各フラグに1を加えたり引いたりといった演算処理はしていま せん。

しかし、Cy マラグだけは他のフラグと達って、強制的に 1 をセットされたり 0 をセットされたりします。また、桁あよれ、パーパーフロー) を超こしたビ ットのデータを著えたり、 据えたデータを状の演纂で視用レジスタに加えたり もします。条件側筋の目的としてよりは、汎用レジスタから桁あよれを起こし た1 ピットを蓄える場所として使用される方が多いくらいです。したがって、 Cy フラグは、本来の機能とレジスタに近い機能の両方を持っていると言うこと ができます。

そこで本書では、Cy フラグの接種を機能をわかりやすくするために、CY レスタという 仮想のレジスタを考えました。 I ビットデータの需えや、 著えた データが利用される場合は CY レジスタとして、 本来の機能である条件判断のために利用される場合は Cy フラグとして扱っており、同じキャリー・フラグを使い方によって 2 つに分けています。たとえば、 Aレジスタを右へ1 ビットシフト (動かす) する場合を考えると、 図5-2のようになります。

図5-3 CYレジスタ(仮種)

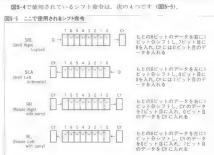


CY レジスタに 1 が蓄えられた結果, Cy フラグに 1 がセットされる

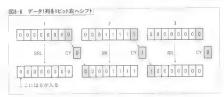
第5章 スムーズ・スクロール

では、CY レジスタにデータが保存されるようすを図5-4で説明しましょう。





この4つのシフト命令を使用すれば、ヘリコアターをスムーズに動かすこと ができます。これらを使って、ヘリコアターのデータ1列を1ビット右へシフ トしましょう(図5-6)、これで令体が1ビット右へシフトしたことになります。

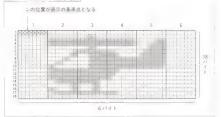


ヘリコプターを横にスムーズに動かす

ヘリコプターを表示させるには

横方向へのスムーズな動きの原理は、シフト命令を利用してドット単位でデータを移動させます。しかし、VRAMの関係上、シフト命令だけでは済みません。そこで、もう一度、リコブターの表示方法について説明しておくことにします。関5-7を見てください。

図5-7 ヘリコプターの表示方法



このへリコブターのキャラクタは■6パイト、縦18パイトで構成されています。 ペリコブターを設計するために、HLペアレジスタを VRAMのポインタ (アドレスの管理)として使用します、HLペアレジスタをは客に象権のの指すが けを管理しており、この基準点をもとに DISPというサブルーチンがヘリコブ ターのデータを DE ペアシスタで読み出してきて、横6パイト縦18パイトの計 18パイトを表示します。

■自由に移動できるのは、左右8ドットまで

ここで基準点を変化させずに左右に移動できるのは、8ドットまでです。したがって、8ドットまでは自由にドット単位で移動できますが、それ以上はドット単位で移動できません。もし、そのまま移動してしまうと、1ドット単位でハリコプターのキャラクタが指えていきます。

9ドット目をどう移動させるか

31 71 4 6 6 7 19 30 6 6 60

そこで、9ドット11の移動は少しや一かいになります。このままでは9ドット目の移動ができないので、ボインタ(HL ペアレジスタのデータ)を1 つ適め 、淡のアドレスから淡点するようにします。しかし、そのままのデータでは 在右どちらかに8ドットずれているので8ドット分ずれて表示することにな り、スムーズに移動できません。そこで、8ドットずれたデータを"山もとに 戻してから、1ドットずらして表示することにします。こうすれば、問題なく 移動することができます。

もう少し具体的な説明をしましょう。 右へ棒動する例を考えます。 ヘリコブ クーのデータは、同転翼を同転させるために、3 つのデータに分けてあります。 この3 つのデータを右へドット単位でシフトすれば、8 ドットまでは問題なく 移動できますが9 ドットHからが問題です。 図5-8で順を迫ってみましょう。

図5-8 9ドット目からの証酬

ここのアドレスを EDDDH 番地とする

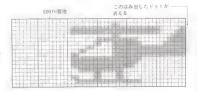


基準のヘリコブターキャラクタ

第5章 スムーズ・スクロール



右へ8ドット移動したヘリコブターのキャラクタ、ここまでは問題なくドット 単位で移動できる。



9ドット移動すると、右側の酸のドットが消えてしまう。 ポインタを1つ進めない限り、右へシフトするとシフトするたびにはみ出した ドットが消えてしまう。

E000H香地



1回でこの8ドット分移動してしまう。

そこで②の状態のデータをそのままにして、ポインタ (VRAM のアドレス)を 1 つ進める E、バイト単位で移動するので 1 回で 8 ドット移動してしまう。



名 カードットシフトドルと、シュラビッドット日の移動位置となり、連続して かいているように登ると

この図でわかるように、9ドット以上を移動させるためには

と繰り返せば良いのです。左へスムーズに移動させる方法も右と全く同じなの

で、特に説明しません。

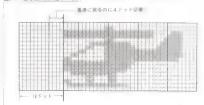
途中で方向を変えて移動するとき

これで、右と左へスムーズに移動できるようになりましたが、まだ問題が残っています。右へ移動していて途中で左に移動するような場合はどうしたら良いのでしょうか。ヘリコプターのデータが基準に戻っていれば問題ないのです

第5章 スムーズ・スクロール

が、図5-9のように途中だったらどうするのでしょうか。

図5-9 基準点に戻る途中のとき



この位置から左端へ移動するドット数は12ドットとなります。移動する左右 のドット数を正確にカウントしていないと、ペリコプターの基準のデータに収 ることができません。そこで、左右へ移動したドット数をカウントするプログ ラムが必要で、そのデータを確保しておかなければなりません。図5-10を使っ て説明しましょう。

図5-10 途中で方向を変え

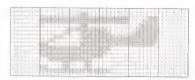


左端に来たとき

左へ移動するためにはポインタのデータから 1 を引いて、基準のデータまで戻してやる必要がある

カウンタのデータ=00(00H)

右端へ移動するのに、ポインタのデータを変えずに16ドット分移動できる。



基準のデータ……カウンタのデータ=08



右端に来たとき

右へ移動するためには、ポインタのデータに!を加えて基準のデータまで戻してやる必要がある。

カウンタのデータ=16(10H)

左端へ移動するのにポインタのデータを変えずに16ドット分移動できる。

■1ドットシフトするたびにカウンタのデータを調べればよい

図5-10を見ると、ポインタ (VRAM のアドレス) を次のアドレスにするか、 ひとつ手前のアドレスにするかはカウンタのデータを参照すれば良いことがわ かります。

つまり、カウンタのデータが 0 や16以外のときは、ボインタをいじらずにた 右へデータをシフトすれば良いし、6 し 0 であれば、データを 8 ドットシフト してボインタを 1 つ暖らせば良いことになります。したがって、1 ドット統右 とちらかにシフトするごとに、カウンタのデータを調べれば良いのです。

スムーズな移動のプログラム

でけ ヘリコプターを左右へスムーズに移動させるプログラムをリスト5-1に

示します。プログラムは第4章のリスト4-5(P.185)と基本的には同じです。 · 部のキールーチン処理と横のシフトルーチンが加わっただけですので、左右の シフトルーチンを中心に説明していきます。

■キールーチンは少し簡単に

今まではた右への移動は、HL ベアレジスタ(VRAM のアドレス、つまりボ インタ)を土1することによって行いました。ですから、たとえば有へ移動す る場合は、HL ペアレジスタに 1 を加えてメイン・プログラムに戻れば良いだけ でした。もちろん、右へ移動した回数をカウントしているDレジスタにも1を 加えなければなりませんでした。

今回はその代りに、右へ1ドットすつ移動するルーチンを呼ぶので少し簡単

こなっています。 ちょっと較べて	みましょう。
今までのルーチン例	今回のルーチン例
RIGHT: LD A, D	RIGHT: LD A, D
CP 4AH	CP 4AH
JP Z, KIYPOP	JP Z, KIYPOP
INC HL	CALL RSM · · ·
INC D	JP KIYPOP
JP KIYPOP	(右へ1ドットずつ移動) するルーチンを呼ぶ

見てわかるように今回の方が少し簡単になっています。RSM ルーチンや LSM ルーチンで、HL ベアレジスタにいつ1を加えたり引いたりするのかを計算し ています.

■左右ヘドット単位で移動するルーチンは

次に、左右へドット単位で移動するルーチンを説明しましょう。主に右への ドット移動のフローチャートを説明します。だへのドット移動のフローチャー トは右とほとんど同じなので、大きく違う点だけを説明します。右へのドット 移動のフローチャートが理解できれば、左へのフローチャートもすぐに理解で きることと思います。

RSM	
((SP))←HL	ポインタ(VRAM のアドレス)をスタックに保存する
HL←SSCD	HLベアレジスタに、左右へ移動したドット数が保存してある アドレス (SSCD) をロードする
A←(HL)	Aレジスタに移動したドット数を入れる
A-16	16ドット移動したか(右端まできたか)
YES A = 16	(O (RSMI) まだ右へ移動できるので、右へドット単位で移動するルーチンへ飛ぶ
((SP))←DE	DE ベアレジスタをスタックに保存
D←-D8H	データを左へ8ドット分移動するための回数をセット (データを基本データに戻すため)
HL←CDATA1	HLベアレジスタに、データ1のアドレスをセットする
LRSM	左へ1ドット移動するサブルーチンを呼ぶ
HL←CDATA2	HLベアレジスタに、データ 2 のアドレスをセットする
LRSM	左へ1ドット移動するサブルーチンを呼ぶ
HL←CDATA3	HL ベアレジスタに、データ 3 のアドレスをセットする
LRSM	左へ1ドット移動するサブルーチンを呼ぶ
D←D-1	Dレジスタから 1 を引く
VES D=0	
NO] DE←((SP))	もとのデータに戻す
HL←((SP))	もとのデータに戻す
D←D+1	Dレジスタに 1 を加える(右へ 1 番地進んだことをカウント



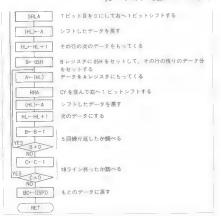
HL ペアレジスタにデータ1のアドレスをセットする HI ← CDATA1



図5-13 右へ1ドット ますョサブルーチン

RET

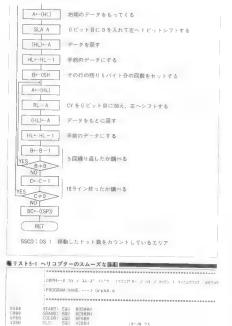




左へ移動するフローチャートは一部を除いて同じなので、大きく違うところ だけを説明します。



第5章 スムーズ・スクロール



120-24 52

```
MON:
                            BE626H
885C
               GRAMB: EQU
               GRAMI: EQU
                            5DH
RRSE
               GRAM2: EQU
               MRAM:
                      EQU
                            5FH
                       ORG START
                       LD (SPSAVE).SP
5988 ED73A68C
8984 11E48C
B082 811058
                            BC.SELSH
B9@A 3E81
898C 3282E6
898F 3E19
8911 3283E6
                            18E683H1.A
B914 3FE8
                            A. RESH
                            (BEGB4H).A
8919 3E88
                      LD
                            A.86H
BRIE SEFE
                      LD
                            A. SEEH
B920 32B9E6
                      LD
                           (0E689H).A
                      CALL COLOR
8926 CD8B42
                       CALL CLS
8929 3E98
               MAIN:
                            A.88R
                            (SSCD).A
B92B 32618B
B92E 2188C8
B931 118688
                            HL. GRAMS
                      1.D
                            DE. BBBBH
                                                : n5" x2 Eado" (7
B934 3E27
                       £D.
               ML1:
                       INC
                            HL
                       INC
8938 3D
B939 C23689
                            NZ.MLI
                            A.100
BC.0050H
B941 89
              MU2:
                      ADD
                            HL.BC
                       INC
8943 3D
                       DEC
8944 C241B9
                            NZ.ML2
8947 8683
                            В.03Н
8949 78
B94A FE83
B94C C25DB9
B94F E5
                           83H
                            NZ,CD2
                      PUSH HL
                      LD DE CDATAL
                      CALL DISP
                      DEC
                      PDP HI
895A C38AB9
                      JP WALT
8950 78
                      LD
                          A.B
895E FE82
                      CP
                      JP NZ.CO3
B968 C271B9
                      PUSH HL
8963 E5
8964 D5
8965 11CEBB
                      LD DE.CDATA2
8968 CD888A
                      CALL DISP
                      DEC
896C DI
                           DE
                      JP
                            WAIT
```

第5歳 スムーズ・スクロール

	B971 8972		CD3:	PUSH PUSH		
	8973 8976	113ABC CD88BA		CALL	DE.CDATAS DISP	
		CD79BA		CALL	WAT	
	897C			90P	BO	
	897D	EI		POP	HL	
	897E 897F	65		PUSH		
	0000	Do CEDO		LO	DE, CDATA2	
		CD88BA			DISP	
		8683		LD	B.03H	
		N1			DE	
	6989	El		POP	HI.	
	0303	61				
	898A	CD79BA	WALT:	CALL	VAT	
	2020	C39EB9		JP	K18	
	Dano	COOLDO				
	B998	D889	WKIY:	EN	A. (09H)	
	8992	E601		AND	BIH	
	8994	C249B9		JP	NZ.KAITEN	
	B997	ED7BA68C		LD	SP. (SPSAVE)	
	0000	C326E6		JP	ANTE	
	рээр	COZOEO		W.	NO.11	
	899E	CS	KIY:	PUSH	RC .	
		DB88		IN	A. (80H)	- 11
	B9A1	47		LD	B.A	
4		E682		AND	82H	
	B9A4	CAD789		JP	Z.LDOWN	
		78		LD	A.B	1.2
		E684		AND		
	B9AA	CAFØB9		JP	Z.DOWN	
	BRAD	77.00		I.D	A.8	
		E688		AND	88H	1.3
		CARRBA		JP	Z. RDOWN	
	Dabo	CHUUDA			Littooan	
	B9B3			LD	A.B	:4
	B9B4	E618		AND	IBH	
	8986	CA19BA		JP	Z.LEFT	
	B9B9	78		LD		: 6
ı		E648 CA25BA		AND	Z,RIGHT	
ı	Dabe	CHESSH		0.5	Z,RIGHI	
	B9BF	7.6		LD	A.B	
		E680		AND	888	
	8902	CA31BA		JP	Z.LUP	
		DB91		1 N	A.(81H)	: 6
	B9C7	5001		LD	B.A	
	8908	E681 CA4CBA		AND	BIH Z.UP	
	Back	CAACBA		JP.	Z.UP	
	BBCD	28		LD	A.B	:9
		E682		AND	62H	
		CA5EBA		JP	Z.RUP	
	8903	CI	KIYPOP	POP	BC.	
	nnn -	googno.		TD		
ž	B9D4	C39889		ar.	MKIY	

§2 ヘリコプターを横にスムーズに動かす

```
B907 7B
              LDOWN: LD
                          A.F
                          RBSH
                      JP
BODE FEED
                           668
                      JP
BRES C5
                      PUSH BC
                      LD BC, 8050H
B9E4 815888
B9E7 89
                           HL.BC
                      CALL LSM
B9F8 7B
B9F1 FEB5
                           BB5H
B9F3 CAD389
                      JP
B9F6 C5
B9F7 815888
                           BC. 0050H
                           BC
BBFC 1C
                      INC
BA60 7B
                           8B5H
BA86 7A
                      JP
BARC CS
SA18 89
BAIL CL
BA12 1C
                      INC
BAI3 CDA7BA
                      CALL RSM
BA16 C3D389
               LEFT:
BA19 7A
BAIA FERR
                           1080
                      JP
BAIC CADSB9
BAIF CD028B
BA22 C3D3B9
                      CALL LSM
BA26 FE4A
                      JP
                      CALL RSW
8A31 7B
               LUP:
BA32 FEBS
                      JP
                            RAH
                      PUSH 8C
BA3E 815000
BA41 A7
BA42 ED42
                      SRC
                           HL.BC
RA44 CI
                      DEC
BA46 CD0288
                      CALL LSM
BA4C 7B
BA4D FE89
                      CP BBH
                      JP
                      JP Z.KIYPOP
PUSH 8C
 BA4F CAD3B9
```

第5章 スムーズ・スクロール

BAS3 \$15000 BAS5 677 BAS5 ED42 BAS5 ED42 BAS5 LD BAS5 LD BAS5 LD BAS5 CAD399 BAS5 CAD399	10	Art. RRSRH	
BASS A7	AND	A	
DAST EDGE	200	90	
BASA ID	DEC	F	
8A58 C3D389	JP	KIYPOP	
BASE 7B	RUP: LD	A.E	
BASE FERR	CP	88H	
BAGI CADJB9	JP	Z,KIYPOP	
DAGE PELA	CD	AAU	
BAGS CARSES	10	7 KIVEGE	
BASA CS	piicu	D. C.	
BASB 015000	1.0	BC. RWSRH	
BAGE A7	AND	A	
BAGF ED42	SBC	HL.BC	
BA71 CI	POP	BC	
BA72 1D	DEC	E	
BA73 CDA7BA	CALL	RSM	
BELP CIDIBR	JP .	KITPUP	
BA79 D5 4	WAT: PUSH	DE	: 5 " カレーマチ
BA7A 1682	LD	D.02H	
BA7C LEAF	⊌2: LD	E. ØAFH	
BATE ID V	#1: DEC	E	
DAYF CZYEBA	950	NZ, WI	
BARS C27CBA	10	NZ MA	
BASS III	PDP	DE	
BA87 C9	RET		
0.110.00			
BASE F3	015P: 01		12°97 / Eads' N-95 99a5 80
BABS USSE	DUCH	IGRAM21.A	199- 7 40 9-0-5
BASC SEL2	1.036	C. 12H	
BASE 6666 1	082: 1.D	B. 86H	
BASS IA I	.OP1: LD	A. (DE)	
BA91 77	LD	(HL1.A	
BA92 23	INC	HL	
BA93 13	INC	DE	
DASA BS	DEC.	NE LODI	
BASS CESTON	Pilsi	BC LOFT	
BA99 014A68	LD	BC, BB4AH	
BASC 89	ADD	HL.BC	
BA9D CI	POP	BC	
BASE SD	DEC	C	
BASE CZEEBA	JP	NZ,LDP2	
BAA2 DOEE	PUP	EMDOM) A	1.174 150 - 11
BAAS FR	EI	CHEMINA	: Dink DV
	-		17726 011
BAAG C9	RET		
BAA7 E5 F	RSM: PUSH	HL	:RIGHT BIT
BAAS 2161BB	LU	RL.SSCD	
BARB /E	CD	A, (HL)	
BANE CZDARA	ID.	N7 DED1	
BAB1 D5	PUSH	DE	
BAB2 1688	LD	D.88B	
BAB4 21628B F	RSM1: LD	HL.CDATA1	
BAB7 CD4488			
HABA 21CEBB	CALL	LRSM	
DARD CDAADD	CALL	HL.CDATA2	
BABD CD44BB	CALL LD CALL	HL.CDATA2 LRSM	:2*37 / ta95* A-9> 9932 E 139- / 40 9*9-5

```
CALL LESM

BACT C28HBA DF NZ.RSMI
BACA D1 POP DE NZ.RSMI
BACB E1 POP HL
BACB E1 NC POP HL
BACB E1 RSMI
     | BAD4 34 | RSRI: INC (HL) | BAD5 25628B | LD | HL.COATAL RSSM BAD8 C1CE8B | LD | HL.COATAL RSSM BAD8 C1CE8B | LD | HL.COATA2 | LD | HL.COATA3 | BAE7 E13ABC | LD | HL.COATA3 | LD | HL.COAT
                                                                                                            CALL RRSM
LD HL.CDATA3
CALL RRSW
POP HL
                                                                     RRSM: PUSH BC
LD C.18
RBA1: LD A.(HL)
       BAE9 C5
       BAEA 8E12
BAEC 7E
BAED CB3F
                                                                   RDAT: LD A, (HL), A
INC HL
LD B, 85H
RBB1: LD A, (HL)
         BAEF 77
       BAFE 23
BAFI 9685
BAF3 7E
       BAF4 CB1F
BAF6 77
                                                                                                                      LD (HL),A
       BAF7 23
BAF8 05
                                                                                                                      DEC
                                                                                                                                             NZ - ROBI
         BAF9 C2F3BA
                                                                                                                    JP
                                                                                                                      DEC
         BAFC 8D
         BAFD CZECBA
                                                                                                                                               NZ.RBA1
                                                                                                                      POP BC
         8888 C1
8881 C9
                                                                                                                RET
         BB1B 213ABC
881E CDE9BA
BB21 15
                                                                                                                  LD HL,CDATA3
CALL RRSM
DEC D
                                                                                                                  JP NZ.LSMI
POP DE
POP BL
           BB22 C20FB8
         8825 DI
8826 EI
         8B26 E1
BB27 2B
         8827 28
8828 15
8829 3E88
8829 326188
882E C9
                                                                                                                    DEC HL
                                                                                                                  DEC N
                                                                                                              LD A, 88H
LD (SSCD)
RET
```

第5章 スムーズ・スクロール

```
8842 E1
                          HI.
BB43 C9
BB44 C5
               LRSM:
                      PUSH BC
 8845 816B88
                           BC.187
BB48 89
                           HL.8C
8849 BE12
8B4B 7E
               LBA1:
                           A. (HL)
BB4C CB27
                           (HL).A
                      DEC
                           HL
8850 0605
BB52 7E
               LBB1:
                           A. (HL)
8853 CB17
                      RL
BB55 77
                           THL), A
BB56 2B
BB58 C252BB
                           NZ, LBB1
BB5C C24BBB
                           NZ.LBAI
BBG2 0000000 CDATAI:DB 00H.60H.60H.60H.60H.00H
BB68 69861869
                         88H.88H.18H.88K.88H.88K
BB6C 9888
BBGE DEFFFFF
                    DB
                          68H. BEEK. BEEH. BEEH. BEH. BRH
8872 8888
8874 88FFFFF
                    DB
                         88H.8FFH.8FFH.8FFH.87H.88H
BB78 8788
BB7A 68861886
                    DB 88H.88H.18H.88H.8FH.88H
BBSD 0001FFC0
                    DB BON. 01H, SFFH, 8C8H, 1FH, 08H
BB84 1F88
BB86 00031FE0
                           00H.03H.1FH.0E0H.3FH.00H
BB8A 3F88
BB8C 888C1FFF
                    DB
                          88H. BCH. 1FH. 8FFH. 8FFH. 8BH
BB92 88381FFF
                          BBH.38H.1FH.8FFH.8FH.88H
                    D8
                         80H. SCBR. 15H. 8FFH. 8FEH. 8BH
BB9C FE88
BB9E 88FFFFE8
                    DB
                          00H.0FFH.0FFH.0E8H.00K.00H
BBA2 8880
BBA4 BEFFFFCE
                    DB
                         88H.8FFH.8FFH.8C8H.88H.B8H
BBAS 6886
BBAA 80FFFF80
                    DB
                          BOH. OFFH. OFFH. 88H. 88H. 88H.
BBAE 8988
BBBB 683FFC86
                    DB
                         88H.3FH.0FCH.88H.88H.88H
8886 00030C00
                          08H.03H.0CH.02H.0DH.00H.00H
BBBC BUCSUCBU
                          BEH. BC3H. BCH. BBH. BBH. BBH
BBC2 BB7FFFE0
                    800
                          88H.7FH.8FFH.8E8H.88H.88H
                          авн. век. еен. вен. изн. вен
                     DB
BBCE ####### CDATA2: DB
```

BBD2 9000

§2 ヘリコプターを横にスムーズに動かす

```
BBD4 88861888
                  DB 86H,88H,18H,86H,88H,88H
BBDS 9988
BBDA 8886FF86
                  DB
                       88H.88H.8FFR.88K.89H.88H
BBDE 9968
BBED BBBBFFB6
                  DB
                       88H.88H.8FFK.88H.87K.89H
BBE4 8788
BBE6 88881888
                  DB
                       884,884,184,884,864,884
BBEA 9F89
BREC SABIEFCS
                  DB
                       88H, 81H, 8FFH, 8C8H, 1FH, 88H
BBF2 88831FE8
                  DB
                       60H.83H.1FH.6E0H.3FH.80H
BBF6 3F88
BBF8 808C1FFF
                  DB
                       88H. BCH. 1FH. 8FFH. BFFH. 88H
BBFC FERR
SBFE 99391FFF
                  DB 00H.30R.1FH.0FFH.0FFH.00H
BC84 8BC81FFF
                  DB BBH. BCBH. LEH. BEEH. BEEH. BBH
BC08 FE09
BCOA OOFFFFEO
                  DB
                       86H.8FFH.8FFH.8E0H.88H.88H
BCI0 08FFFFC0
                  DB
                        BEH. BFFH. BFFH. BCBH. BBH. DBH
BC14 8888
BCI6 ØSFFFFB0
                  0B
                       38H.8FFH.8FFH.80H.88H.88H
BC1C 803FFC80
                       80H.3FH.0FCH.00H.00H.00H
BC22 99938C89
                       88H.83H.8CH.89H.88H.88H
                  98
BC26 8888
BC28 88C38C88
                  DB
                       884.8C3H.BCH.88H.88H.88H
BC2E #87FFFE8
                  ES
                        88H.7FH.8FFH.8E8H.88H.88H
BC32 8608
BC34 BERRERER
                  88 88H. 86H. 86H. 88H. 68K. 66H
BC38 8888
BC3A 86888888 CDATA3: DB 88H,88H,88H,88H,88H,88H
BC3E 8688
BC48 98881888
                  DB
                       39H. 86H. 18H. 88H. 88H. 86H
BC46 08001800
                  D6 88H.88H.18H.88H.88H.88H
BC4A 8888
BC4C 08081808
                  DB 00H.00H.18H.00H.07H.00H
                       88H.88H.18H.88H.9FH.86H
BC52 00001800
                  DB
                       88H.81H.8FFH.SC8H.1FH.88H
BC5C 1FB8
BCSE 00031FE0
                       88H,83H,1FH,8E8H,3FH,88H
BC64 888CIFFF
                  DB
                       88H. 8CH. 1FH. 8FFH. 8FFH. 88K
                  DB
                       88H. 38H. LEH. REEH. REEH. REH
BC70 08CBIFFF
BC74 FE00
                        88H. SCSH. 1FH. SEEH. SEEN. SSH
BC76 02FFFFE0
                  DB
                       gall accu accu acau gau gau
                        89H.8FFH.8FFH.8C8H.88H.88H
BC88 8888
BC82 00FFFF80
                  DB 88H.8FFH.8FFH.88H.88H.88H
BC86 8698
BCSS 083FFC00
                  DB
                        een. 3FH. RFCH. BeH. BeH. BBH
BCSC 8888
BORE ARRONCAR
                  DB 88H, 93H, 9CH, 88H, 88H, 88H
BC94 08C30C00
                  DB
                        68H, 6C3H, 8CH, 88H, 88H, 88H
BC9A 087FFFE0
                  DB
                       88H.7FH.8FFH.8E8H.88H.88H.88H
```

第5章 スムーズ・スクロール

BCA9	8689 8689 8689 8689	DB	88H,88H,88H,88H,88H,88H
BCA6 BCA8 BCE4		SPSAVE: DS STACK:	82H 68
BCE4		ENI	D

テキスト画面とグラフィックス画面を重ね合わせる

PC-8801mkIISR は、テキスト画前とグラフィックス画面の重ね合わせが簡 単です 図5-15のような画面を作ってみましょう。

テキスト文字は8ビット×8ビットで構成されているので、縦の 8ビットで構成されているので、縦の 8ビット分がグラフィックス両面 め近なります。グラフィックス両面 面がテキストの文字までいかない ように、グラフィックス両面の のドット数を数えているEレジス 夕から単ビット引いておきます。 テキスト文字のカラー化は第2章 (P.93)でやっていますので思い 出してください、テキスト両面と



グラフィックス両館の取ね合わせのプログラムをリスト5-2に示します。リスト 5-1にテキスト文字を表示する部分を追加しただけです。テキストモードを同時 に使用することができるので、ゲームなどのスコアやメッセージを表示する場 合に便利です。

●リスト5-2 テキスト画面との重ね合わせ

	************	古代女女女女女女女女女女女女女女女女女女女女女女
	GRPH9 30 / 24-2" 1	トゥウ ト テキスト カッメン ノ カサネアワセ
	PROGRAM NAME> Gr	ph9.e

BSBB	START: EDU 88908H	
CSSS	GRAMS: EQU BCBBBH	
6F6B	COLOR: EQU 6F6BH	
428B	CLS: EQU 428BH	:カーツル ケス

§2 ヘリコプターを横にスムーズに動かす

E626 005C 805D 005E 005F		MON: GRAME: GRAMI: GRAM2: MRAM:	EQU	ØE626H 5CH 5DH 5EH 5FH	
		;	ORG	START	
	ED73DEBC 311CBD			(SPSAVE), SP SP.STACK	: 29+7 / 91t
898A 898C 898F 8911 8914 8916 8919 8918 8928	811958 3E81 32B2E6 3E19 32B3E6 3EE8 32B4E6 3E88 32B4E6 3EFF 32B9E6 CD686F		LD LD LD LD LD LD LD LD	BC.5819H A.81H (8E6B2H).A A.19H (8E6B3H).A A.88E8H (8E6B4H).A A.88FH (8E6B8H).A A.8FFH (8E6B9H).A COLOR	
B926	CD8B42		CALL	CLS	
892D	DD2118F4 DD360888		LD	[X.0F418H ([X+0].00H	: テキスト モシ゛ ノ ヒョウシ゛
	DD3681C8	;		(1X+1),8C8H	; F#XF J-35- #40
8938 8938	21C1BC 11E0F3 011C00 EDB0		LD	HL.TEDATA DE.BF3E0H BC.28	; デキスト ノ ヒヨウシ
		:			
	3E88 327CBB	MAIN:		(SSCD),A	
B948 B94B	2188C8 118888 3E27		LD	HL.GRAMS DE.8888H A.278	: N5" X/ E395" (F
B953	14 3D C24DB9 3E64	ML1:	INC DEC JP LD	HL D A NZ.ML1 A.180	
B958 B959 B95A	1C	ML2:	ADD	BC.8058H HL.BC E A NZ.ML2	
B95E	7B D508		LD	A.E 88H E.A	: TEXT / ト゚ット プン とク
B962	0603		LD	8.83H	
B965	78 FE03 C278B9	KAITEN	CP JP	A.B B3H NZ,CD2	

第5章 スムーズ・スクロール

```
PUSH HL
B96A E5
                    PUSH DE
896B D5
896C 117DBB
                    LD DE.CDATA1
B96F CDA3BA
                    CALL DISP
B972 85
                        В
                        DE
                    POP HL
8974 E1
                    JP WAIT
8975 C3A5B9
B978 78
              CD2:
                    LD
                    CP 62H
JP NZ,CD3
8979 FE02
B978 C28CB9
B97E E5
                    PUSH HL
B97F D5
B988 11E9BB
                    LD DE.CDATA2
                    CALL DISP
                    DEC B
                    POP
                        DE
8987 DI
                    POP HL
B988 E1
                    JP
                        WAIT
B989 C3A5B9
B98C E5
              CD3: PUSH HL
898D D5
                    PUSH DE
                    LD DE.CDATAS
B98E 1155BC
B991 CDA3BA
B994 CD94BA
                    CALL WAT
                    POP DE
8997 D1
                    POP HL
8998 E1
                    PUSH HL
B99A D5
                    PUSH DE
                    LD DE.CDATA2
899B 11E9BB
                    CALL DISP
BSSE CDASBA
                    POP DE
POP HL
B9A1 8683
B9A5 CD948A
              WAIT: CALL WAT
                    JP KIY
B9A8 C3B9B9
              WKIY:
                    IN A. (R9H)
B9AB DB89
                    AND 81H
89AD E681
89AF C264B9
                         NZ.KAITEN
B9B2 ED7BDEBC
                         SP. (SPSAVE)
                     .IP
                         MON
B9B6 C326E6
                    PUSH BC
C5
                    IN A. (80H)
B9BA DB00
B9BC 47
                         B.A
B9BD E682
                    AND 82H
B9BF CAF2B9
                    JP
                    LD A.B
B9C2 78
                    AND 04H
89C3 E684
B9C5 CAØBBA
                    JP Z.DOWN
                    LD A.B
B9C8 78
B9C9 E608
                    AND '88H
```

§2 ヘリコプターを横にスムーズに動かす

				9 2	ヘリコフターを横に
BSCB	CAIBBA		JP	Z.RDOWN	
Bace	78 E610	1	LD AND	A.B 10H	: 4
B9D1	CA34BA		JP	Z.LEFT	
B9D5	78 E548 CA48BA			A.B 40H Z.RIGHT	; 6
B9DA B9DB B9DD	78 EG88 CA4CBA		LD AND JP	A.B 80H Z.LUP	17
89E6 89E2 89E3 89E5	47		IN LD AND JP	A.(81H) B.A 81H Z.UP	: 8
	78 E682 CA79BA		AND	A.B B2H Z.RUP	:9
BSEE	C1	KIYPOP	POP	BC	
BSEF	СЗАВВЭ		JP	WKIY	
B9F5 B9F8 B9F9 B9FE B9FF BA62 BA63 BA64 BA65 BA68	FEAE CAEEB9 7A FE00 CAEEB9 C5 015000 III C1 IC CD1DBB C3EEB9	LDOWN:	CP JP LD CP JP PUSH LD ADD POP INC CALL JP	BC.8058H HL.BC BC E LSM KIYPOP	
BA9C BA9E BA11 BA12 BA15 BA16 BA17 BA18 BA18 BA1B BA1E BA21 BA21 BA22	815880 89 C1 1C C3EEB9 78 FEAE CAEEB9 7A FE4A CAEEB9	DOWN:	LD CP JP	BC.0050H HL.BC BC E KIYPOP A.E 0AEH Z.KIYPOP A.D 4AH Z.KIYPOP	

第5章 スムーズ・スクロー

BA28 89 BA20 01 BA20 10 BA2E 01 BA31 03			ADD POP INC CALL JP	BC.8850H HL.BC BC E RSM KIYPOP
BA34 7A BA35 FE BA37 CA BA3A CE BA3D CE	E00 AEEB9 DIDBB	LEFT:	LD CP JP CALL JP	A.D 00H Z.KIYPOP LSM KIYPOP
BA48 7/ BA41 FE BA43 C/ BA46 CE BA49 C3	E4A AEEB9 DC2BA	RIGHT:	CP JP	A.D 4AH Z.KIYPOP RSM KIYPOP
BA4C 7E BA4D FE BA4F CA BA52 7A BA53 CA BA53 CA BA56 CB BA56 CA BA50 EL BA5C CA BA5D EL BA61 CL BA61 CL BA64 CC	E68 AEE89 AEE89 5 15008 7 042	LUP:	LD CP JP LD CP JP PUSH LD AND SBC POP DEC CALL JP	A.E 88H Z.KIYPOP 86H Z.KIYPOP EC BC,S8H A HL.BC BC EC LSM KIYPOP
BA67 76 BA68 FF BA6A CC BA6E B BA71 A BA72 EF BA74 C BA75 11 BA76 CC	E00 AEEB9 5 15000 7 D42	ŪP:	LD CP JP PUSH LD AND SBC POP DEC JP	A.E 00H Z.KIYPOP BC BC.0050H A HL.BC BC E KIYPOP
BA7C C BA7F 7 BA88 F BA82 C BA85 C BA86 B BA89 A	E88 AEEB9 A E4A AEEB9 5 15888 7 D42 1 D DC2BA	RUP:	LD CP JP LD CP JP PUSH LD AND SBC POP DEC CALL JP	A.E 88H Z.KIYPOP A.D 4AH Z.KIYPOP BC BC,8858H A HL.BC BC E RSM KIYPOP

§2 ヘリコプターを横にスムーズに動かす

```
(シュカン マチ
BAA1 base of the property of t
```

第5世 スムーズ・スクロール

```
BAF6 21E9BB
                           LD HL.CDATA2
CALL RRSM
BAF9 CD84B8
BAFC 2155BC
BAFF CD84BB
                           LD HL.CDATA3
### RET | FOP HL |
### RET |
### LD C.1
### LD 6.1
### BB88 CB3F
### BB88 CB3F
### BB88 77
                            LD C.18
LD A.(HL)
                             LD (HL), A
                             INC HL.
 3BØB 23
888C CBIF RB): LD A.(HL)
                           LD (HL), A
1NC HL
DEC B
 3B11 77
 BB12 23
BB13 05
                             JP NZ. RBB1
 BB14 C28EBB
                        DEC
JP
                            DEC C
JP NZ.RBA1
POP BC
 BB17 8D
 8818 C207BB
 BB1B C1
                          RET
 BBIC C9
:LEFT BIT
 BB3D C22ABB
                                   NZ.LSM1
                             JP
                            POP DE
POP HL
 BB49 D1
 BB41 E1
                          DEC HL
DEC D
LD A.88H
LD (SSCD).A
 BB42 2B
 BB43 15
BB44 3E88
BB46 327CBB
 BB49 C9
                             PET
 BB4A 35 LSR1: DEC (HL)
BB4B 217DBB LD HL, CDATA1
BB4E CDSFBB CALL LRSM
BB51 21E9BB LD HL, CDATA2
CALL LRSM
CALL LRSM
CALL LRSM
CALL LRSM
CALL LRSM
 BB51 21E9BB
BB54 CD5FBB
BB57 2155BC
BB5A CD5FBB
BB5D E1
                             LD HL.CDATA3
                             CALL LRSM
                             POP HL
                             RET
BBSE C9
BBSF C5
                   LRSM: PUSH BC
 BB60 816B80 LD BC,107
 BB63 89
```

```
6864 ØE12
BB66 7E
             1.BA1:
                         A. (HL)
BB67 CB27
                    SLA A
BB69 77
                    LD
                         (HL),A
BB6A 2B
                    DEC HL
BB68 8685
                         B. 05H
             LBB1:
                         A. (HL)
BBGD 7E
BB6E CB17
BB70 77
                    LD
                         (HLI.A
                         HL
BB71 2B
                    DEC
BB72 05
BB73 C26DBB
                    .1P
                         NZ.LBB1
BB76 ØD
                    DEC
BB77 C266BB
                    JP
                         NZ.LBAI
                    PDP
BB7B C9
                    RET
BB7C
             SSCD: DS
                                          1220-1 how 20
BB7D 8888888 CDATA1:DB 88H, 88H, 88H, 88H, 88H, 88H, 88H
BB81 8888
BB83 00001800
                  BBH.00H.18K.00H.00H.00H
BBB9 ØØFFFFFF
                  DB 88H.8FFH.8FFH.8FFH.88H.88H
BBSD BBBS
BBSF BEFFFFFF
                  DB 88H, 8FFH, 8FFH, 8FFH, 87H, 88H
BB93 8788
BB95 00001800
                  DB 08H,08H,18H,08H,8FH,89H
BB99 9F88
BB9B 0001FFC8
                  DB 88H.81H.8FFH.8C8H.1FH.88H
BRSE LERR
BBA1 00031FE0
                   DB 60H.03H.1FH.8E8K.3FH.80H
BBA5 3F86
BBA7 908C1FFF
                  DB 00H, 0CH, 1FH, 0FFH, 0FFH, 00H
BBAB FFRB
BBAD 00301FFF
                  D8 86H.36H.1FH.8FFH.8FFH.88H
BBB1 FFBB
BBB3 08C01FFF
                  DB 88H, BC8H, 1FH, 8FFH, 8FFH, 88H
BBB7 FEBB
BBB9 00FFFFE0
                  DB 08H, 8FFH, 9FFH, 8E8H, 38H, 80H
BBBD 0000
BBBF ØRFFFFCA
                  DB 00H.8FFH.0FFH.8C8H.86H.86H
BBC3 8888
BBC5 #0FFFF80
                  DB 88H.8FFH.8FFH.80H.86H.88H
BBC9 0000
BBCB 003FFC00
                  DB 08H.3FH.0FCH.89H.88H.08H
BBCF 0000
BBD1 00030C00
                  DB 98H,83H,8CH,88H,88H,89H
BBD5 0000
BBD7 80C38C89
                  DB 08H.8C3H.8CH.80H.80H.88H
BBDB 0000
BBDD 867FFFE8
                  DB 02H.7FH.8FFH.8E0H.80H.00H
BBE1 6686
BBE3 80088008
                  DB 08H,88H,88H,08H,88H,88H
BBE7 0000
BBE9 00000000 CDATA2:DB 00H.00H.00H.00H.00H.00H
```

BBED	9889		
	80811888	DB	80H, 80H, 18H, 80H, 80H, 80H
BBF3			
	6880FF86	DB	99H. 88H. 8FFH. 88H. 88H. 88H
BBF9	8880FF88	DB	88H,88H,8FFH,08H,97H,88H
SBFF			
BC91	80081800	DB	08H.00H.18H.09H.0FH.00H
BC85		DD.	88H.81H,8FFH.8C8H.1FH.88H
	8001FFC0 1F00	UD	BBN. BIN, BFFN. BCDN. IFN, BBN
	00031FE0	DB	88H.83H.1FH.8E8H.3FH.88H
BC11	3F80		
	880C1FFF	DB	88H.8CH.1FH.8FFH.8FFH.88H
	FF00 00301FFF	DB	00H.30H.1FH.8FFH.8FFH.00H
	FF88		
BC1F	80CB1FFF	DB	80H, 8C0H, LFH, 6FFH, 8FEH, 80H
BC23	FE00		88H.8FFH.8FFH.8E8H.88H.83H
BC25	00FFFFE0	1000	UER. BFFH. BFFR. BCBR. GDR. GDR
	BOFFFFCO	DB	00H.0FFH.0FFH.0C8H.08H.08H
BC2F	6889		
	00FFFF80	DB	00H.0FFH.0FFH.80H.00H.00H
	8888 883FFC88	700	889.3FH.8FCH.88H.88H.88H
	8888	-	Beat of the City Control of the City Control
	00030C00	DB	00H.03H.2CH.88H.88H.88H
BC41	9689		
	80C30C80	DB	80H.8C3H.8CH.80H.88H.80H
	8998 887FFFE8	DB	08H,7FH.0FFH,0E0H.00H.00H
	9888		
	800000000	DB	98H.08H.08H.09H.00H.00H
BC53	9999		
8055	22222222	CDATA3: DB	88H.88H,88H,88H.88H.88H
BC59	8888		
	00801800	DB	08H,82H,18H,00H,00H,00H
	9999	80	99H.00H.18H.89H.09H.98H
	8808	UB	98H. 00H. 15H. 00H 00H 00H
BC67	66816866	DB	89H.88H,18H,88H.87H.80H
	8780		
	00001800	DB	96H.00H.18H.88H.9FH.89H
	8981FFC8	DB	88H.81H.8FFH.8C8H.1FH.88H
	1F88	DD	S S S S S S S S S S S S S S S S S S S
	00031FE0	DB	88H.83H.1FH.8E8H.3FH.88H
BC7D	3F88		
BC7F	000CIFFF FF00	DB	08H.8CH.1FH.8FFH.8FFH.00H
	00381FFF	DB	88H.38H.1FH.8FFH.8FFH.80H
BC89	FF00		
	80C01FFF	DB	88H.8C8H.1FH.8FFH.8FEH.88H
	FE00 00FFFFE0	DB	88H.8FFH.8FFH.8E8H.88H.88H
BC95		UU	
BC97	00FFFFC0	DB	89H.8FFH.8FFH.8C8H.89H.80H
BC9B	0000		

```
BC9D 00FFFF80
                     DB BEH. BFFH. BFFH. 88H. 88H. 88H
BCA1 0000
BCA3 803FFC88
                     DB
                          BAH. SER. RECH. RAH. RAH. BAH.
BCA7 8888
BCA9 00030C00
                     DB
                          80H. 93H. BCH. 88H. 68H. 88H
BCAD 8898
BCAF 00C30C00
                     DB 88H, 9C3H, 9CH, 98H, 98H, 98H
BCB5 007FFFE0
                     DB 98H.7FH.9FFH.9E9H.89H.88H
BCB9 8868
всве есеевее
                           BRH. BRH. BRH. BRH. BRH. BRH
BCBF BBBB
BCC1 47525848 TEDATA: DB 47H, 52H, 58H, 48H, 28H
BCC5 26
BCC6 54455854
                     DB 54H, 45H, 58H, 54H, 28H, 4DH, 49H, 58H, 28H
BCCA 264D4958
BCCE 20
BCCF 44454D4F
                     DB 44H, 45H, 4DH, 4FH, 28H, 58H, 52H, 4FH
BCD3 2050524F
BCD7 4752414D
                           47H.52H.41H.4DH.88H.88H.88H.88H
RCDE
              SPSAVE: DS 82H
                         60
BDIC
```

DMAを止めて処理スピードを上げる

リスト5-1、5-2ではアログラムのアルゴリスムの関係でマシン語でも動きが 非常に謹くなっていました。そこで、プログラムのアルゴリズムはそのままで スピードを上げてみましょう。PC-8801シリーズで処理スピードが遅くなって いる原因は CRTC(xPD3301)へのデータ転送です。これはテキストモードの VRAM を読み込むのに DMA (Direct Memory Access) 処理をしているため に、CPUがストップきせられてしまうためです。もし、DMA 処理をしていな ければ30%以上のスピードアップをはかることができます。ただし、テキスト 両面を使用する場合にこのままでは使用できません。

> LD A, DIH LD (OE6B2H), A

■5章 スムーズ・スクロール

- LD A. 19H
- LD (DE6B3H), A
- LD A, 0E8H
- LD (0E6B4H), A
- LD A, 00H
- LD (0E6B8H), A
- LD A, OFFH
- LD (OE6B9H), A

この2つのプログラムを初めと終りの方に入れるだけですから楽です。実際 ゲームなどを待る場合に表示する文学からすべてグラフィックスを健用した方 が出来上がったときの評価は高いですから。テキスト文字を使用しない方が繋 明です。そのときにこのプログラムが彼こまつでしょう。

この章で作ったリスト5-1に追加をして処理スピードを上げてみました。今度 は速すぎて回転費のタイミングがとれません。WAIT ルーチンのデータを変え てスピードを遅くしてください。

では、プログラムをリスト5-3に示します。

これで第5章を終わります。グラフィックスの基本的なことは説明しました。 この他に細かいことはたくさんありますが、今までの部分で説明できなかった ことは次の「マシン語アラカルト」で説明していきたいと思います。

■リスト5-3 スムーズな移動(DMA ストップ)■

**********		****		*****	*****	***	
GRPH+-8 30					ノ カイテン	トキーニュウリョク	88
PROGRAM NAI	Æ> G	£38.e 1	DIVA STI	327			
********		******	3 34 00 00 00 00	****	***	*******	
CR#0 6F6B 428B E626 005C 005D	GRAMS: EGCOLOR: EGCLS: EGMON: EGGRAM8: EGGRAM1: EGGRAM2:	QU 8890 QU 6F61 QU 4281 QU 8E6. QU 5CH QU 5CH QU 5CH QU 5FH	96H BH BH		: 'n-У	L 52	
		RG STA	RT				
8900 ED73C9BC 8904 3107BD	: U		SAVE).S	SP	: 29%	ク ノ タイ ヒ	

```
LD BC.5819H
8907 011950
B90A 3E01
                     LD A.BIH
B90C 32B2E6
                         (@E6B2H).A
898F 3E19
                          A.19H
                          (@E6B3H).A
B911 32B3E6
                          A-BESH
8916 3284E6
                          (@E6B4H).A
                     LD
                          A.88H
B919 3E88
                         (BE6B8H).A
B91B 32B8E6
                     LD
B91E 3EFF
8928 32B9E6
                     1.0
                         (BESB9H).A
                     CALL COLOR
8923 CD6B6F
B926 CD8B42
                     CALL CLS
B929 3E00
                     LD A.80H
OUT (51H).A
                                             :DMA STOP | 過加した部分
8928 D351
892D 3E88
892F 3284BB
              MAIN:
                      LD A.08H
                      LD (SSCD).A
                      LD
                          HL. GRAMS
B932 2180C0
                                              : ハシニメノ ヒョウシニイチ
B935 118808
                      LD DE.0000H
                          A.27H
B938 3E27
             ML1:
                      INC
                          HL
B93A 23
B93B 14
                      INC
                      DEC
B93C 3D
                           Α
893D C23AB9
                      39
                           NZ.MLI
B940 3E64
                           A.108
                           BC.0050H
8942 015000
             ML2:
                          HL . BC
B945 89
B946 1C
                      INC
                           E
8947 3D
                          A
8948 C245B9
                      JP.
                           NZ.ML2
8948 9693
                      1.D
                           B.03H
B94D 78
B94E FE03
              KAITEN: LD
                          A.B
                     CP
                          03H
                     JP
B956 C261B9
                          NZ. CD2
B953 E5
                     PUSH HL.
B954 D5
                     PUSH DE
                      LD
                           DE-CDATAI
B958 CDABBA
                     CALL DISP
B958 85
                     DEC
                          10
B95C D1
                          DF
B950 E1
                         HL.
                     POP
B95E C38EB9
                     JP.
                          WAIT
8961 78
                          A.B
8962 FER2
                          82H
B964 C275B9
                     JP
8967 E5
                     PUSH HL
B968 D5
B969 | 11F1BB
                     LD
                          DE.CDATA2
B96C CDABBA
B96F 05
                     CALL DISP
                     DEC
                          В
B978 D1
                     POP
                          DE
                          HL
B971 E1
                     POP
B972 C38EB9
                     JP.
                          WAIT
```

```
B975 E5
              CD3:
                    PUSH HL.
 8976 D5
                     PUSH DE
B977 115DBC
B97A CDABBA
B97D CD9CBA
898B D1
                    LD DE.CDATA3
CALL DISP
                    CALL WAT
POP DE
                    POP HL
B981 E1
                   PUSH R
 B983 III
 8984 11F1BB
                   LD DE.CDATA2
B987 CDABBA
                    CALL DISP
B98A 0603
                    LD B.83H
POP DE
                     POP HL
              WAIT: CALL WAT
 B9BE CD9CBA
 B991 C3C1B9
                     JP KIY
8994 DB09 WKIY: IN A.(09H)
B996 E691
                    AND RIH
B998 C24DB9
                    JP NZ.KAITEN
B99B 611950
                   LD BC.5819H
                                      :DMA START
 899E 3E01
                   LD A.81H
LD (8E6B2H).A
LD A.19H
 89AØ 32B2E6
 B9A5 32B3E6
                   LD IBEGBSHI,A
                                                     温加した
部分
B9A8 3EE8
                   LD A. 6E8H
B9AA 32B4E6
                    LD (8E6B4H),A
B9AD 3E00
B9AF 32B8E6
                    LD A. 88H
                    LD (RESBSH).A
 B9B2 3EFF
                    LD
B9B4 32B9E6
                        (SEGBSH),A
 B987 CD6B6F
                   CALL COLOR
B9BA ED7BC9BC
                 LD SP (SPSAVE)
                        MON
 39BF C326E6
 B9C1 C5
             KIY: PUSH BC
 89C2 DB66
                     IN A. (88H)
 B9C4 47
                         8.A
 89C5 E692
                    AND 02H
 B9C7 CAFAB9
                     JP Z.LDOWN
 BBCA 78
                    LD A.B
                   AND 64H
 BBCB E684
 B9CD CA13BA
                     JP Z.DOWN
 B9D8 78
                    LD A.B
 B9D1 E508
                    AND 08H
 B9D3 CA23BA
                     JP Z.RDOWN
                    LD A.B
 B9D6 78
 B9D7 E618
                    AND 18H
                     JP Z.LEFT
 воро сазсва
                     ID A.E.
                                            : 6
89DC 78
230
```

```
B9DD £648
                     AND 48H
B9DF CA48BA
                     JP
                          2.RIGHT
B9E2 78
                     LD
                          A.B
B9E3 E688
                     AND 89H
B9E5 CA54BA
                     JP
BSE8 DB81
                     1N
                           A. (8)H1
                                           :8
B9EA 47
                     LD
                          B-A
B9EB E601
                     AND BIH
B9ED CA6FBA
                     JP
B9F8 78
                                               :9
B9F1 E602
                     AND 82H
B9F3 CA81BA
                     JP
BSF6 C1
              KIYPOP: POP BC
B9F7 C394B9
                      JP
                          WKIY
B9FA 7B
              LDOWN: LD
                          A.E
B9FB FEB5
                          ØB5H
B9FD CAF6B9
                          Z.KIYPOP
BARR 7A
BARI FERR
                     LD
                          BBH
BA03 CAF6B9
                     JP
                     PUSH BC
LD BC.8059H
BA86 C5
BA87 015888
                          HL.BC
BARA 09
                     ADD
                          BC
BABB CI
                     POP
BARC 1C
BARD CD25BB
BAIR C3F6BB
                     INC
                     CALL LSM
                     JP KIYPOP
BA13 7B
              DOWN:
                          A.E
BA14 FEB5
                         885Н
                     JP Z.KIYPOP
                     PUSH BC
LD BC.0858H
BAIA 015000
BAID 89
                           HL.BC
BATE CI
                          BC
BAIF IC
                     INC
BA28 C3F6B9
                      JP
              RDOWN:
                     LD
BA23 78
                           A.E
BA24 FEB5
                     CP
                          BBSH
                     JP
BA26 CAF6B9
BA29 7A
                           A.D
BAZA FE4A
                           46H
BA2C CAF6B9
                     JP
                           Z.KIYPOP
BA2F C5
                     PUSH BC
BA36 815088
                     LD
                           BC.0050H
BA33 09
                     ADD
                           HL.BC
BA34 C1
                           BC
BA35 1C
                     INC
BA36 CDCABA
                     CALL RSM
BA39 C3F6B9
                     JP KIYPOP
BA3C 74
```

第5章 スムーズ・スクロール

BA3F BA42	FE00 CAF689 CD2588 C3F689		CP JP CALL JP	Z.KIYPOP	
BA4B BA4E	7A FE4A CAF6B9 CDCABA C3F6B9	RIGHT:	CP JP CALL	A.D 4AH Z.KIYPOP RSM KIYPOP	
BA57 BA58 BA58 BA60 BA61 BA64 BA65 BA67 BA68 BA69	FE00 CAF689 7A FE00 CAF689 C5 015880 A7 ED42 C1	LUP:	LD CP JP LD CP JP PUSH LD AND SBC POP DEC CALL JP	BC.50H A HL.BC BC E LSM	
BA72 BA76 BA76 BA79 BA7A BA7C BA7D	FE88 CAF6B9 C5 B15889 A7 ED42 C1	UP:	LD CP JP PUSH LD AND SBC POP DEC JP	08H Z.KIYPOP BC BC.0050H A HL.BC BC	
BA84 BA88 BA88 BA8D BA8E BA91 BA92 BA94 BA95 BA96	FE08 CAF6B9 7A FE4A CAF6B9 C5 015080 A7 ED42 C1	RUP:	LD CP JP LD CP JP PUSH LD AND SBC POP DEC CALL JP	BBH Z.KIYPOP A.D 4AH Z.KIYPOP	
BASF BAA1 BAA2 BAA5	1682 1EAF 1D C2A1BA 15 C29FBA	VAT:	PUSH LD LD DEC JP DEC JP POP	DE D.02H E.0AFH E NZ.W1 D NZ.W2 DE	; シ`カン マチ

BAAA C9	:	RET	
BAAC D35E BAAE C5 BAAF WE12 BAB1 0606 BAB3 1A BAB4 77 BAB5 23 BAB7 05 BAB6 C2B3BA BAB8 C5 BAB6 C2B3BA BAB8 C5 BAB6 C1 BAC2 C2B1BA BAC2 C2B1BA BAC5 C1 BACC C2B1BA BAC5 C1 BAC6 D35F	LOP2: LOP1:	OUT (GRAMAY: A PUSH BC LD C. 12H LD B. 60H LD A. 10E LD B. 60H LD A. 10E LD A. 10E LD A. 10E LD A. 10E LD BC LD BC LD BC LD BC LD BC. 604 AH ADD HL. BC POP BC LD BC CALLOP2 FOP BC CALLOP2 FOR BC CALLOP	: タッラフ / セェウシ・ ルータン ブリコミ NO: カラー / 4D タッサーン 1カラー / 4D タッサーン 1メイン メモリー こ スル
BAC8 FB		EI	;77⊐E OK
BAC9 C9	1	RET	
BACA E5 BACB 2184BB BACE 7E BACF FE10 BAD1 C2F7BA BAD4 D5 BAD5 1608 BAD7 2185BB BADA CD67BB BADA CD67BB BADB 21F1BB BAEB CD87BB BAEB 3215DE BAEB CD67BB	RSMI:	LD HL.SSCD LD A.(HL) CP 16 3P NZ.RSR1 PUSH DE LD D.08H LD HL.CDATA1 CALL LRSM	:RIGHT BIT
BAEG CD67BB BAE9 C2D7BA BAED D1 BAEE B1 BAEF 14 BAF8 23 BAF1 3E88 BAF3 3284BB BAF6 C9		CALL LRSM LD HL.CDATAS CALL LRSM DEC D JP NZ.RSM1 POP DE POP HL INC HL LD A.28H LD A.28H LD A.88H RET	
BAF7 34 BAF8 2185BB BAFB CD0CBB BAFE 21F1BB BB01 CD0CBB BB04 2150BC BB07 CD0CBB BB0A E1 BB0B C9	RSR1:	INC (HL) LD HL.CDATA1 CALL RRSM LD HL.CDATA2 CALL RRSM LD HL.CDATA3 CALL RRSM POP HL RET	
BBBC C5	RRSM:	PUSH BC	

```
BB@D @E12
                                                 RBAL: LD C.18
     BB0F 7E
    BBIR CB3F
                                                                       SRL
    BB12 77
                                                                                       (HL) .A
    BB13 23
                                                                       INC
                                                                                    HI.
  BB14 0685
                                                                                    B. 85H
   8816 7E
8817 CB1F
                                                RBB1:
                                                                       LD
                                                                                      A. (HL.)
                                                                       RR
    BB19 77
                                                                                      (HL),A
    BB1A 23
                                                                       INC
                                                                                     HL
    BB18 85
                                                                                     В
   BB1C C2168B
BB1F 0D
                                                                       JP
                                                                                      NZ.RBB1
                                                                       DEC
    BB2B C28FB8
                                                                      JP
                                                                                      NZ - RBAI
                                                                       POP BC
    BB24 C9
   BB25 ES
                                                LSM: PUSH HL
                                                                                                                                             :LEFT BIT
                                                 LD HL.SSCD
   BB26 2184BB
   BB29 7E
BB2A FE00
                                                                  CP ##
JP NZ.LSR1
    BB2C C252BB
   BB2F D5
BB30 1608
   BB30 1608 LD D.08H
BB32 2185BB LSM1: LD HL.CDATAI
   BB3E 215DBC
                                                               LD HL.CDATA3
CALL RRSM
DEC III
   BB41 CD0CBB
  BB44 15
  BB45 C232BB
                                                                    JP NZ.LSMI
  BB48 DI
                                                                   POP DE
  8849 F1
                                                                 POP HL
DEC HL
  BB4A 2B
  BB4B 15
BB4C 3E08
BB4E 3284BB
                                                                  DEC III
| BB4C | 2348B | BB51 | C9 | RE5 | BB52 | C5 | LSR1: DEC | (HL) | BB52 | C5 | LSR3: DEC | 
                                                                   LD A.08H
  BB59 21F1BB
BB5C CD67BB
BB5F 215DBC
BB62 CD67BB
                                                                  LD HL.CDATA3
CALL LRSM
  BB65 E1
                                                                     POP HL
  BB66 C9
  BB67 CH LRSM: PUSH BC
  BB68 016B00
                                                                  LD BC.187
  BB6B 89
BB6C ØE12
BB6E 7E
                                                                     ADD HL.BC
                                          LD C.18
LBA1: LD A.(HL)
 BB6F CB27
BB71 77
BB72 28
BB73 8685
                                                                       SLA A
                                                                      LD (HL).A
                                      DEC
LD
LBB1: LD
                                                                                  H1.
                                                                    LD B.85H
  BB75 7E
                                                                                     A. (HL)
 BB76 CB17
                                                                      RL
                                                                                     A
```

```
BB78 77
BB79 2B
                          (HL).A
                     DEC
                          HL
BB7A 05
                     DEC
BB7B C275BB
                     JP
                          NZ.LBB1
BB7F ØD
                     DEC
BB7F C26EBB
                         NZ.LBA1
BB82 C1
BB83 C9
                     RET
                                             : 72D-1 17v1 20
BB85 88088888 CDATA1:DB 88H.88H.88H.88K.88H.88H.88H
BB8B 88801808
                   DB 888,888,188,888,888,888
BBSF 8888
8891 MMFFFFFF
                   DB 88H.8EEH.8EEH.8EEH.88H.88H
BB97 ØØFFFFFF
                   DB 08H, 8FFH, 8FFH, 8FFH, 87H, 80H
BB9D 66881886
                   DB 08H, 08H, 18H, 08H, 0FH, 88H
BBAI 0F00
BBA3 0001FFC0
                   DB 00H,01H,0FFH,0C0H,1FH,00H
BBAS DEB31FEE
                   DB 08H, 03H, 1FH, 0F0H, 3FH, 00H
BBAD 3F86
BBAF 608C1FFF
                   DB 08H-8CH-1FH-8FFH-8FFH-80H
BBB5 00301FFF
                   DB 08H.38H.1FH.8FFH.8FFH.89H
BBB9 FF88
BBBB 00C01FFF
                   DB BBH, BCBH, 16H, BEEH, BEEH, BBB
BBC1 ØOFFFFE0
                   DB 68H.8FFH.8FFH.8E8H.89H.68H
BBC5 8888
BBC7 ##FFFFC0
                   DB ARH OFFH AFFH OCRE ORE ACH
BBCB 8896
BBCD ARFFFFRR
                   DB 88H. 8FFH. 9FFH. 88H. 88H. 88H. 88H
BBD1 6680
BBD3 003FFC00
                   DR 96H 3EH 6ECH 99H 86H 86H
BBD7 9889
BBD9 80830C88
                   DB 80H.83H.8CH.80H.88H.80H
BBDD 8688
BRDE BRC38C88
                   DB 88H AC3H ACH 88H 88H 88H 88H
BBE3 8988
BBES 007FFFE0
                   DB 88H.7FH.8FFH.8E8H.89H.89H
BBES SARS
ваев направан
                   DB 88H.88H.88H.88K.88H.88H
BBEF 0000
BBF1 68968968 CDATA2: DB 89H.88H.88H.88K.88H.88H
BBF5 8888
BBF7 88861888
                   DB 39H, 68H, 18H, 69H, 68H, 88H
BBFB BBBB
BBFD 8888FF88
                   DB 88H.88H.8FFH.88H.88H.88H
                   DB 80H.08H.0FFH.00H.87H.08H
BC03 0000FF00
BC87 8788
                   DB 80H.86H.18H.89H.8FK.8FH
```

BCBD			
	8881FFC8	DB	80H.01H.0FFH.0C8H.1FH.08H
		DB	00H.03H.1FH.0E0H.3FH.00H
		DE	80H.8CH.1FH.8FFH.8FFH.2GH
		DE	DON. DCM. IFM. DFFM. DMFM. DDM
BC21	80381FFF	DB	69H.38H.1FH.0FFH.0FFH.68H
	FFBB		
	SECRIFFF	DB	68H. BCBH. 1FH. BFFH. BFEH. 80H
BC2B	FE06	nn.	241 45511 25511 25411 2411 2211
		DB	08H.8FFH.0FFH.8E8H.08H.88H
		DR	00H.0FFH.0FFH.0C0H.08H.08H
8037		VII	DBH. BFFH. BFFH. CCDII. CDII. CDII.
BC39		DB	08H.8FFH.8FFH.80H.80H.80H
BC3D	0000		
		DB	88H,3FH.8FCH.80H.8BH.88H
		100	88H.83H.9CH.88H.88H.88H
		-	88H.8C3H.8CH.88H.86H.80H
		000	BEN. BCJN. ECH. BBN. BBN. BBN
		DB	88H.7FH.8FFH.0E8H.08H.08H
BC55	9889		
BC57	99999999	100	80K. 90H. 80H. 80H. 88H. 86K
BC58	9699		
OCCD	10000000	ED + 2 + 0 + 00	88H.884.88H.88H.88H.88H
		CDATAS - DD	005.025.004.004.004.004
		DB	88H.88H.16H.88H.86H.88H
BC67	8888		
BC69	88881888	DB	88H. 98H. 18H. 88H. 88H. 88H
		DB	80H.88H.18H.98H.07H.88H
			ODA. UUA. IOA. DBA. D/A. DBA
	9789		
BC75	98881889		88H.88H.18H.88H.8FH.36H
BC75 BC79	98881888 9F00	DB	88H.88H.18H.88H.8FH.88H
BC75 BC79	98881889	DB	
BC75 BC79 BC7B BC7F BC81	00001800 0F00 0001FFC0 1F00 00031FE0	DB DB	88H.88H.18H.88H.8FH.88H
BC75 BC79 BC7B BC7F BC81 BC85	00001800 0F00 0B01FFC0 1F00 00031FE0 3F00	DB DB	88H.88H.18H.88H.8FH.36H 88H.81H.8FFH.9C8H.1FH.86H 88H.83H.1FH.9E8H.3FK.86H
BC75 BC79 BC7B BC7F BC81 BC85 BC87	00001880 0F00 0001FFC0 1F00 00031FE0 3F00 000C1FFF	DB DB	08H.80H.18H.88H.8FH.06H 08H.81H.8FFH.9C8H.1FH.08H
BC75 BC79 BC7B BC7F BC81 BC85 BC87 BC88	00231800 0F00 0B91FFC0 1F00 0B931FE0 3F00 0B0C1FFF	DB DB DB	08H.88H.18H.88H.8FH.08H 08H.81H.8FFH.0C9H.1FH.08H 08H.83H.1FH.0E9H.3FK.06H 08H.8CH.1FH.8FFH.8FH.88H
BC75 BC79 BC7B BC7F BC81 BC85 BC87 BC8B BC8D	00201800 0F00 0B01FFC0 1F00 00031FE0 3F00 080C1FFF FF00 00301FFF	DB DB DB	88H.88H.18H.88H.8FH.36H 88H.81H.8FFH.9C8H.1FH.86H 88H.83H.1FH.9E8H.3FK.86H
BC75 BC79 BC7B BC7F BC81 BC85 BC87 BC8B BC8D BC91 BC93	00001888 0F00 0801FFC0 1F00 08031FE0 3F00 080C1FFF FF00 080C1FFF 080C1FFF	DB DB DB	08H.88H.18H.88H.8FH.36H 08H.81H.8FFH.9C9H.1FH.98H 08H.83H.1FH.9E8H.3FK.86H 08H.8CH.1FH.8FFH.8FH.88H
BC75 BC79 BC7B BC7F BC81 BC85 BC87 BC8B BC8D BC91 BC93 BC97	00001800 0F00 0B01FFC0 1F00 08031FE0 3F00 080C1FFF FF00 00301FFF FF00 080C1FFF	DB DB DB DB DB	88H.88H.18H.8FH.38H 88H.81H.8FFH.9C8H.1FH.88H 88H.83H.1FH.8EBH.3FH.88H 88H.8CH.1FH.8FFH.8FFH.88H 88H.38H,1FH.8FFH.8FFH.88H 88H.38H,1FH.8FFH.8FFH.8FH
BC75 BC79 BC78 BC7F BC81 BC85 BC87 BC88 BC8D BC91 BC93 BC97 BC99	00001888 0F00 0801FFC0 1F00 08031FE0 3F00 080C1FFF FF00 08301FFF FF00 08C01FFF FF00 08CFFFE0 08FFFFE0	DB DB DB DB DB	06H. 88H. 18H. 88H. 86H. 36H 08H. 81H. 8FFH. 0C9H. 1FH. 08H 06H. 83H. 1FH. 0E8H. 3FH. 86H 08H. 8CH. 1FH. 8FFH. 8FFH. 88H 80H. 30H. 1FH. 0FFH. 8FFH. 88H
BC75 BC79 BC78 BC7F BC81 BC85 BC87 BC88 BC8D BC91 BC93 BC97 BC99 BC90	9881888 960 9881FFC8 1F00 98831FE0 3F80 888C1FFF FF08 98C01FFF FE08 88CFFFE8 88FFFFE8	DB DB DB DB DB DB	08%, 88%, 18%, 89%, 89%, 89%, 89%, 89%, 89%, 89%, 8
BC75 BC79 BC78 BC7F BC81 BC85 BC87 BC8B BC8D BC91 BC91 BC92 BC92 BC95 BC95 BC96 BC96 BC97	9881888 960 9881FFC8 1F90 98931FE6 3F90 98931FFF FF90 88301FFF FF90 88C81FFF FE00 88FFFFE8 88FFFFE8	DB DB DB DB DB DB	88H.88H.18H.8FH.38H 88H.81H.8FFH.9C8H.1FH.88H 88H.83H.1FH.8EBH.3FH.88H 88H.8CH.1FH.8FFH.8FFH.88H 88H.38H,1FH.8FFH.8FFH.88H 88H.38H,1FH.8FFH.8FFH.8FH
BC75 BC79 BC78 BC7F BC81 BC85 BC87 BC88 BC80 BC91 BC91 BC92 BC99 BC99 BC9D BC9F BCA3	98881888 9700 9881FFC0 1F90 88831FE0 3F80 88801FFF FF88 88301FFF FF88 88281FFF FF88 88C81FFF 88C8 89FFFFC0 8907FFFC0	DB DB DB DB DB DB DB	08H.88H.18H.8FH.08H 88H.81H.8FFH.0C9H.1FH.08H 08H.08H.1FH.9EFH.3FH.88H 08H.8CH.1FH.0FFH.8FFH.89H 08H.3CH.1FH.0FFH.8FFH.08H 08H.3CH.1FH.0FFH.0FFH.08H 08H.6CFH.1FH.0FFH.08H.08H.08H
BC75 BC79 BC78 BC7F BC81 BC87 BC88 BC87 BC98 BC99 BC99 BC99 BC99 BC95 BC95 BC95 BC95	08881888 0F00 08831FFC0 1F00 08031FE8 3F00 080C1FFF FF08 08301FFF FF08 08C81FFF FE00 08FFFFE8 00FFFFC8	DB DB DB DB DB DB	08%, 88%, 18%, 89%, 89%, 89%, 89%, 89%, 89%, 89%, 8
BC75 BC79 BC78 BC7F BC81 BC85 BC87 BC88 BC89 BC91 BC93 BC99 BC99 BC95 BC95 BC95 BC96 BC96 BC96 BC96 BC96 BC96 BC96 BC96	08881888 0F00 0881FFC0 1F00 08831FE8 3F80 088C1FFFF08 08301FFF F80 08261FFFFE8 080FFFFE8 080FFFFE8 0808 080FFFFE8	DB DB DB DB DB DB DB DB	08H.88H.18H.8FH.08H 88H.81H.8FFH.0C9H.1FH.08H 08H.08H.1FH.9EFH.3FH.88H 08H.8CH.1FH.0FFH.8FFH.89H 08H.3CH.1FH.0FFH.8FFH.08H 08H.3CH.1FH.0FFH.0FFH.08H 08H.6CFH.1FH.0FFH.08H.08H.08H
BC75 BC79 BC78 BC87 BC81 BC85 BC87 BC88 BC97 BC99 BC99 BC9F BCA3 BC9F BCA3 BCA5 BCA5 BCA5 BCA6 BCA6 BCA6 BCA6 BCA6 BCA6 BCA6 BCA6	988 3 888 0 600 0	DB DB DB DB DB DB DB DB DB	08M.88M.18M.89M.8FM.08M 08M.81M.9FM.0C9M.1FM.08M 08M.83M.1FM.8FFM.8FFM.88M 08M.30M.1FM.8FFM.8FFM.88M 08M.30M.1FM.8FFM.8FFM.88M 08M.8FFM.8FFM.8FFM.8BM 08M.8FFM.8FFM.8BM.88M 08M.8FFM.8FFM.8BM.88M 08M.8FFM.8FFM.8BM.8BM
BC75 BC79 BC78 BC87 BC81 BC85 BC87 BC88 BC93 BC97 BC99 BC99 BC99 BC99 BC99 BC99 BC99	9828 888 9780 9780 981 1750 982 1750 982 1750 982 1750 982 1750 982	DB DB DB DB DB DB DB DB DB	08H.83H.16H.89H.0FH.30H 88H.81H.0FFH.0CH.1FH.06H 08H.83H.1FH.8FH.3FH.89H 08H.30H.1FH.0FFH.8FH.89H 08H.30H.1FH.0FFH.8FH.89H 08H.30H.1FH.0FFH.0FH.08H 08H.0FFH.8FFH.8FH.08H.08H 08H.0FFH.8FFH.8CH.88H.08H
BC75 BC79 BC78 BC87 BC81 BC85 BC87 BC88 BC91 BC93 BC99 BC99 BC96 BC96 BC96 BC96 BC96 BC96	9883 888 676 676 688 676 688 676 688 676	DB DB DB DB DB DB DB DB DB	08%, 88%, 18%, 89%, 89%, 88%, 89%, 89%, 89%, 89%, 8
BC75 BC79 BC79 BC81 BC85 BC87 BC88 BC8D BC91 BC92 BC99 BC99 BC99 BC99 BC99 BC43 BC45 BC45 BC45 BC45 BC45 BC45 BC45 BC45	98281880 9881FFC0 1790 9881FFC0 1790 9881FFC0 9881FFC0 98201FFF FF00 98201FFF FF00 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF	DB DB DB DB DB DB DB DB DB	08H.88H.18H.8FH.06H 08H.89H.9FH.06H).IFH.08H 08H.89H.1FH.8EBH.3FH.89H 08H.30H.1FH.8FFH.8BH 08H.30H.1FH.8FFH.8FH.8BH 08H.6CBH.1FH.8FFH.8FH.08H 08H.8FFH.8FFH.8FEH.08H 08H.8FFH.8FFH.8GH.8BH.8BH 08H.8FFH.8FFH.8GH.8BH.8BH
BC75 BC79 BC79 BC81 BC85 BC87 BC88 BC80 BC91 BC93 BC99 BC99 BC99 BC99 BC99 BCA3 BCA5 BCA5 BCA5 BCA5 BCA5 BCA5 BCA5 BCA5	98281880 9881FFC0 1790 9881FFC0 1790 9881FFC0 9881FFC0 98201FFF FF00 98201FFF FF00 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF 98201FFF	DB	06M.83H.18H.85H.86H.36H 86H.81H.0FFH.0C9M.1FH.86H 06M.05H.1FH.86FH.3FH.86H 06H.3CH.1FH.0FFH.8FH.86H 06H.3CH.1FH.0FFH.8FH.86H 06H.3CH.1FH.8FFH.8FH.06H 06H.6FFH.8FFH.86H.06H.06H 06H.8FFH.8FFH.8C8H.06H.86H 06H.8FFH.8FFH.86H.86H.06H
	8C13 8C15 8C19 8C19 8C19 8C19 8C19 8C19 8C19 8C19	BC13 1F08 BC15 08031FE8 BC15 08031FE8 BC15 08031FE8 BC25 FE8 BC37 0808 FFF6 BC37 0808 FFF6 BC37 0808 FFF6 BC37 0808 FF6 BC37 0808 FF6 BC37 0808 FF6 BC37 0808 FC8 BC38 BC39 0807 FC8 BC39 0808 BC39 BC39 BC39 BC39 BC39 BC39 BC39 BC39	6C13 1F88 DB 6C15 38831FEB DB 6C15 38831FEB DE 6C15 38831FEB DE 6C17 97886 DB 6C27 96831FEF DB 6C28 96831FEF DB 6C28 96831FEF DB 6C29 96831FF 6C29 96831F 6C29 96831

§2 ヘリコブターを横にスムーズに動かす

United States		6888 6888868 6888		DIE	80H.88H.00K,00K.88K.80H
	BCC9 BCCB BD87		SPSAVE: STACK:	DS DS	82H 68
	BD87			END	



マシン語アラカルト

à la carte 1

()はどういうときに 使うのだろう

①(HL)とHLの違いは? ②(8010H)と8010Hの違いは?

たとえば、ペアレジスタを使用すると きに、

LD HL, 0F3C8H

のように |) がある場合とない場合が あります。この2つはどう違うのか、ま ず考えてみましょう。

■HLと(HL)は違う

ベアレジスタを使用するとき。| 1の あるのとないのでは意味が全く違いま

*****...

■ ()がない■合 ペアレジスタにデータを入れる

例 LD HL, OF3C8H

HL ペアレジスタに F3C8H をセットします(HレジスタにはF3H, LレジスタにはC8H が入ります).

● () がある場合

()の中のペアレジスタに入っている データをアドレス・データとして、その アドレスへデータをロードする

例① LD HL, 0F3C8H ② LD (HL), A

①で HL ペアレジスタに F3C8H がセットされます。②では、HL ペアレジスタを アドレスを示すデータとして扱い、 HL ペアレジスタの指し示す器地へAレジス タのデータを書き込みます。この場合は、 F3C8H 番地にAレジスタのデータが書 き込まれます。

直接アドレッシングと 間接アドレッシング

また、ただ F3C8H 番地へAレジスタ のデータを書き込むだけならば、次のよ うにも書くこともできます。

LD (0F3C8H), A 直接アドレッシング

直接アトレッシン

=LD HL, OF3C8H

LD (HL), A

間接アドレッシング

"LD (457528H)、A" は直接アドレス を指定していますので、直接アドレッシ グラといいます。それに対して、1回 HLペスレジスタをどしてアドレスの値を セッドしてから、メモリに青き込む方法 を間接アドレッシンプといいます。それ ぞれの別途は次のように考えていいでしょう。

直接アドレッシングと 間接アドレッシングの用途

■■酸アドレッシング

LD (ΔΔΔΔΗ), A - バイトや2バイト程度のデータで、決 まった番地でなおかつ連続性がないとき

マルチタスク可能な割り込み機能

に使用する

■国産アドレッシング

LD HL, AAAAH

連続性のある番地で、データ数が多いと を使用する

次に、レジスタでなく数値に () が つく場合を与えてみましょう. たとえば、 LD HL, 8010H

LD HL, (8010H) の2つはどう違うのでしょうか。形が同 じで同じ命令のようにみえますが、実は 全く違うのです

8010Hは数値, (8010H)は番地を表す

① LD HL, 8010H Hレジスタに80H. Lレジスタに10H をセ この間接アドレッシングは、HLペア レジスタを使用していますので、HLペ アレジスタに1を加えたり、1を引いた り、または他のペアレジスタともたし身 引き繋ができますので、その分だけ、ア ドレスを指定するのが寒になります。

ットする

② LD HL, (8010H)

16ビット・データをメモリから転送する 命令、8010H番地のデータをレレジスタ にセットし、801IH番地のデータをHレ ジスタにセットする

①は HL ペアレジスタに直接データ をセットするのに対し、②は () で指 定されるメモリ番地と次の番地にあるデ ータを HL ペアレジスタにセットする 命令です (データは連順にセットされま す).

à la carte 2

マルチタスク可能な割り込み機能

割り込みという機能は一体, どういうもの なのだろうか?

電話のベルが 鳴った

具体的な例をとり説明すると次のよう になるでしょう。 A 代は会社のデスクで ある仕事をしています。そこに電話のベ ルがなりました。電話の相手は上司の部 後です。その部長から、急いでこの仕事 をやってくれとたのまれました。すると A君は今までの仕事を中断して、部長の 仕事を始めます、つまり、A君にとって は別の仕事が割り込まれたことになりま す、A 召は部長の仕事を終えると、6 と の仕事に戻ります。

しかし、A 引は。いつも部長からの仕事をやっていたのでは、本来の自分の仕事ができないのでことわることもできます (実際には無理……)

割り込みを受けたり 拒んだり

激活のベルはA 君に留長の仕事を割り 込ませる舞っかけを作っていますが、こ なりません、A 君が部長の仕事をしたり、 ことわったりするのはソフトウェアでで きます。これは、IFFレジスタをコント ロールすることによって行います

[EII] 命令は別の仕事を引き受けてもよ いですよというものであり、「DII会会は 埋が終ったら知り込みOK じします 今は別の仕事を引き受けることができま

せんという命令です。

割り込み OK!EI 命令 IFF (インタラブト・フリップ・フロ

割り込み不可! …… DI 命令

ップ) をりにする IFF を1にする

PC-8801シリーズでは動血処理にこの 割り込みを使用しているので、グラフィ ックス用のカラーブレーンに書き込むと きは割り込みを禁止してからカラーブレ ーンに書き込まなくてはなりません 気

à la carte 3

先入れ後出し スタック・ポインタ

退避命令などに出てくるSP(スタック・ポイ ンタ)の意味と使い方がわからない。

「PUSH, POP, CALL, RET *Z-80Aの内部レジスタ の命令を使うときには必ずこのスタッ ク・ボインタ (以後 SP と略す! の働きが 重要になってきます。この SP について 説明しましょう

なぜSPが 必要になるのか

まず、どうして、SPが必要なのかを考 えてみましょう、Z-80A の内部レジスタ は右関のようになっています。 汎用レジ スタを中心に話を進めましょう。 HL ペ アレジスタは主に16ビットの溶質ができ るので、番地の間接指定などに多く使用 されます。もし、HLペアレジスタを

Đ′ B′	E'
B	400
	C.
A	F
I	
X	
P	
C	
	X Y P C

VRAM のアドレス指定用に使用してい て、 さらに、別のアドレスを HI. ペアレ ジスタで指定する必要が生じた場合はど うするのでしょう

HLペアレジスタが いっぱいなとき

ば、 - 番簡単なのは DE ベアレジスタや BC ベアレジスタなどに EX 命令で保存 する方法です。また、「EXX 命令」で副 レジスタセットに保存する方法もありま

しかし、もし、D. E. B. Cなど のレジスタを他の目的のために使用し ていたのであれば、この方法はとれませ ん、もし、Aレジスタを使用できるとし たならば、以下のように HL ペアレジス タのデータをメモリに保存することがで きます

● A レジスタを使った HL ペアレジスタ の退避

LD A. H LD (0E500H), A HI ベアレジスをの保存 LD A. L. ID (0E501H), A この間 HL ペアレジスタ は自由に係える LD A. (DE500H)

LD H A HI ベアレジスをもとの データに戻す LD A. (DESOLH) LD L. A

このように、「PUSH, POP 命令」を使 用しなければ、16バイトのメモリを使用 したくてはなりません

SPはレジスタ保存番地の 管理人

やっかいな問題は、保存したメモリの 器康をプログラマ自身が1っかりと管理 しなければならないことです

これは非常にやっかいなことです。た もし、SPを使用できないとしたなら とえば BC, DE, AF ペアレジスタの保存 が生じた場合を考えてください。何邪地 にどのレジスタのデータが保存! てある かを覚えておくのは大変です。これでは、 プログラマに負担がかかりすぎます

> そんなときに、レジスタの保存してい る番地を自動的に管理してくれるのが SP なのです。この SP のおかげで、プロ グラマはどの番地にどのレジスタのデー タが保存されているかを、全く気にせず に済みます

○ SPはどのように 台動(のだろう

この SP を使用するにあたっては、専 用レジスタの SP にどのメモリエリアを 使用するか、アドレスをセットしなけれ ばたりません.

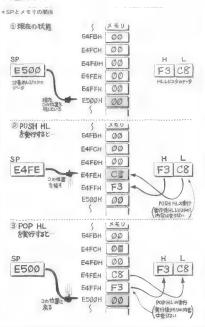
セットするには、

LD SP. DESORH とすれば、SP 専用レジスタに E500日 番

地がセットされます。これで、SP が使用 可能になります。

SPとメモリの 関係

たとえば「PUSH HL! と「POP HL! 命令を使用したときの SP とメモリの関 係を図で説明すると P.244の図のように なります.



このように SP は番地の小さい 方に向 かってカウントしているので、一般的に はメモリのワークエリアの途中にはセッ トせず、終りの方にセットするのが普通 です。

SPエリアの 確保のし方

SPの使用メモリのエリアを確保すると きには次のようにしなければなりません。 たとえば、30レベルのスタック・エリ アを確保するときに必要なバイト数は、

30×2パイト+ 1=61パイト つまり、連続して30回スタック・エリア を使用する場合は肌パイト分必要である ことになります

しかし、実際には、連続して30回データをスタック・ボインタに値むことは、 あまりありません。

の他の使用例 OCALL とRET

「PUSH、POP」命令ばかりでなく。 「CALL、RET」命令でも、SP は使用されています。次の命令の SP の動きを見てみましょう。

メインプログラム サブルーチン

① CALL WAIT → WAIT : LD B, OFH

(2) RET

①CALL WAIT を実行すると、現在 のPC (プログラム・カウンタ)のデータ をSP で退避します。次に WAIT という サブルーチンが入っているアドレスを PC にセットし、実行します。 ② RET 命令があると、SP で混凝して いたデータを PC にセットします、PC の データがもとのデータに戻った。 そこの 番地から実行を開始します。

なお、この PC は分岐命令などがない 限り次の実行すべき番地を指しているの で、戻ってもすぐ実行できるのです。

クリアにも 使える

このように SP は一般的にはレジスタ のデータの保存用に使用されますが、他 の使い 方もあります、SP エリアを VRAMにすれば、これを利用して調面 のクリアができます

また、SPはコンピュータにとって大 事を機能なので、この機能を最大限に生 かすコンピュータもあります、SPをう まく使用すると、ステップ数が短くなる という特長を利用したもので、連ポーラ ンド式による計算方法がそれです。

à la carte 4

いろいろ使える スタック操作命令

スタック操作命令にはどんな種類があり、 どんなときに使われるのだろうか.

スタック操作命令とはスタック・ポイ タック操作命令は4つに分けて考えるこ ンタを使用する命令のことです。このス とができます。

スタック・ポインタでデータをセットす II 命令群 (LD 命令)

LD SP、 OBFFH トタークをセットすることによって、スタック・ポインタを製造

LD SP、(nn) - nn で指定したアドレスから SP にIGビットデータがセットされる

LD (nn), SP ー nn で採定したアトレスに SPのデータが書き込まれる

LD SP, HL

LD SP. IX SPに各16ビットレジス系のデータをセットする

LD SP. IY

EX (SP), HL

EX (SP), IV SPの示すアドレスのデータと各ISビットレジスタのデー目を交換する

EX (SP), IX

これらの命令はSPのデータを見た トする場合や、複数の軽似スタック・ポリ、新たにセットすることができます。 インタを必要とする場合に活用されま新たにスタック・ポインタの保護をセッ す。

命令に関する命令制

ADD HL. SP ーHLペアレジスタとSPとを足し算してHLペアレジスタにセット

ADD IX, SP ーIXインデックス・レジスクとSPとを足し算してHLペアレジスタにセット

ADD IY, SP ーIYインデックス・レジスタと SP とを定し算して HLペアレジスタにセット

ADC HL, SP - ADD HL, SPと同じだが、キャリーも一緒に加える SBC HL, SP - キャリーも含めて HL ペアレジスタから SP を引く

INC SP

⊸ SP C | ಹಸಾಸಿಹ

DEC SP

-- SPから1を引く

レジスタ退避命令 PUSH, POP

レジスタの退避をする場合に使用され、一番多く使用されます。この退避命令ではPUSHとPOPの順番を間違えないようにしないと大変なことになります。

PUSH HL— PUSH BC— PUSH BC— PUSH AF \$ POP AE --

POP BC ←

POP DE ←

POP HI ←

スクック・ポインタでデータを保存す る場合、初めに退選したデータは一番最 後に見ってきます

HLペアレジスタと BCペアレジスタを交換

意図的に順番を要求ると面行いことも を表す、たとえば、HLペアレジスタと BCペアレジスタのデータを受験したい 場合です、残念ながら、IEX HL. BCJ という命令はないので、この順番を意図 的に読えることによって、データを交換 しようとするわけです。

HL ベアレジスタと BC ベアレジスタの 交換

PUSH HL.

PUSH BC.

POP HL ←

POP BC ←

・番最後に「PUSH BC」を使用してス タックにデータを保存し、次の「POP HLJで本来はBCペアレジスタに入るべ きデータをHLペアレジスタに入れて しまうのです、この方法は4パイトで済 みますので、「LDJ命令を使用するよりは るかにプログラムが買くなります

表面に表れない

表面に表れませんが PC (プログラム・ カウンタ)の退避を行う命令があります。 icc「CALL」、「RET」命令群です。

CALL 無条件にコールする RET 無条件に戻る CALL Z, nn Z=1のときコール RET Z Z=1のとき戻る CALL NZ, nn Z=0でコール

RET NZ Z=0で戻る CALL C, nn C=1でコールする RET C C=1で戻る

CALL NCnn C=0でコールする RET NC C=0で戻る CALL M, nn S=1でコールする

RET M S=1で戻る CALL P, nn S=0でコールする

RET P

RET PO

CALL PE, nn P/V=1でコールする RET PE P/V=1で戻る CALL PO. nn P/V=0でコールする

S=0で戻る

P/V=0で戻る

(Z, C, S, P/V はフラグ)特にこの「CALL」命令を多く使用する

特にこの「CALL」命令を多く使用する と、スタックが積まれますので注意が必 要です。

à la carte 5

SPI **COOOH以前に**

マシン語プログラムを作る際。どのように SPを設定したらよいか、

N₈₈-BASIC からマシン語のプログラ てしまいます。そこで新しく SPを設定 ムに入った場合。8レベル(16バイト) 分しかマシン語用スタックエリアが用意 されていません。少し大きなプログラム 前にSPを設定しないといけません。そ になるとこれでは不足してしまい、Nan- こで、BFFOH 帯地に SPを設定してみ BASICのワークエリアのデータを博し

の VRAM のことを考えると、CONOH U ましょう.

ORG DB000H

ID (DRFFAH) SP · No-BASIC の SP の位置を BFFAH 美原と BFFBH 美統に係在する

LD SP、OBFFOH ・マシン請用に 8FFOH 棚間に新しく SP をセットする

LD (OBFFCH)、SP - また、マシン語のプログラムを実行させるときのために、BFFCH、BFFDH 書地にデータを保存しておく(海び、BASIC から入らない棚合は必要ない)

LD SP、(OBFFAH) - もとの SP の位置に戻しておく必要がある。そうしないと無上する

RET ← BASIC に思る JP MON モニクへ戻る

なお、SPを BFF0H 番地に設定して いるので、BFF0H 番地以前の数1-バイ ト分は絶対に使用できません。

また、マシン語のプログラムから

BASIC のプログラムに戻る場合は、必ず もとのSPの位置に戻してからでない と、基业をおこします。

à la carte 6

RSTで 再開始番地を操作

RST 命令がよくわからない。面白い使い方があるのだろうか

RST 命念は再スタート (RESTART) 香地を操作する命令です、Z-80A は外部 的に 再 スタート で き る 香 地 を8+ LNMI) 側もっています。その 8 側の 番 遠は次のようになっています。

亜スタート・アドレス

	1 - 1
0000H	(RST OOH)
H8000	(RST 08H)
0010H	(RST IOH)
H8100	(RST 18H)
0020H	(RST 20H)
0028H	(RST 28H)
0030H	(RST 30H)
0038H	(RST 38H)

割り込み時に参照される 再スタート・アドレス

000日番地はリセットをかけたときにスタートする番地ですから、対別な客地です。他の?側は何故必要なのでしょうか。0008日から0038日までの了側は、ソフトウェア的にどうしても必要なのにはありません。本当に必要な少したシステムで、割り込みをかけるときにどうしても必要な機能です。本実はこの利り込みのとは参照されるアドレスが、この再スタート・アドレスをのです。ソフトウェア的にも使用であるように、RST命会が利用を含れて、

いるのです、PC-8801シリーズでは、モード2という強力な割り込み機能を使用していますので、この再スタート・アドレスは参照していません。 そこで、ソフトウェアだけに使用しています。

モニタに 戻るとき

マシン語のプログラムの終りでモニタ モードに戻る場合は、本書では次のアド レスにジャンプしていました。

JP 0E826H ←本書で使用 この他に,

JP OE669H

としても、モニタへ戻ります。実験して みてください。ジャンプ命令を使用する と3パイトを必要とします。

RSTだと 1パイトで済む

しかし、"RST 38H"はモニタのワー クエリアとなっているために、RST 命令 を使用すると、

RST 38H (マシン語で FFH) で、1バイトで済みます。

モニタモードに戻る3つの方法

① JP 0E828H --- 3 バイト命令 ② JP 0E669H --- 3 バイト命令 ③ RST 38H --- レバイト命令

0038円には 何が書かれているか

このように再スタート・アドレスを参 照するようなときは、IP 命令を使用する よりも、1バイトの RST 命令を使用し た方が側利です それでは、0038日 8世 には何が参かれているのでしょうか。モ ニタで逆アセンブルすると右上のリスト のようになります

"RST 38H"を実行すると、0038H 番

8886	C3 E669	JMP	E669
003B	E88	NOP	
003C	6 B	NOP	
083D	8.0	NOP	
803E	80	NOP	
003F	8.6	NOP	

地へジャンプしてここから実行します。 すると、0038H 番地には「IP 0E669H」 の命令が参いてあります。

```
hide@ge.eggs
E888 88 FF Edie Mil FF de FF de FF de FF de FF de FF de FF
E828 88 FF ERSE BE FF BE
ED40 00 FF E050 00 FF E868 88 FF 88
              FF 88
                     PF 88 FF
                              88 PP
                                    BE EF BE FF BE FF
ENTS BR FF BR FF BR
                     FF 88 FF
                              88 FF 88 FF
                                          88
                                                 88 FF
ENBO FF NO FF NO FF
                     SE EF MB EF
                                BB FF
                                       88 FF
                                             318
ERSO FF NO FF NO FF
                     88 FF 88 HF
                                 SD FF SS FF
                                             BE FF 88
EBAS FF 88 FF
              88 FF
                     88 FF 88 FF 88 FF
                                       88 FF 88 FF 88
E880 FF 80 FF 80 FF 60 FF 80 FF 80 FF 88 FF 80 FF 80
ERCE FF RR FF
              88 FF
                     UN EF NO EF NO EF NO EF
                                                FF OR
ENDS FF 80 FF
              88 FF US FF US FF BE FF BE FF BE FF BE
ERER FF AR FF DO FF OR FF OR FF BR FF BR FF AR FF AR
ENFO FF 80 FF 98 FF 98 FF 80 FF 80 FF 80 FF 88 FF 88
```

この「RST 38H」は実に面白いこと HとFFHになっています。 をしてくれます、とのリストはE000H 番地から EOFF 帯地までのダンプリス トになります.

RST 38HT 暴走を食い止める

このままの状態で E000日 番地から宝

モニタへ戻りましたね。このダンプ上 の番地ならばどこからスタートしても。 モニタに戻ります、PC-8801シリーズは、 電源投入時に使用していないエリアが00

もし、これらの番地にプログラムが暴 走してきたならばどこかで CPU は、OP す。 すると CPU は0038H 番地へ自動的 にジャンプしてしまいます。 そこはモニ タのワークエリアですから、モニタのコ マンドとなり、禁止を食い止めることが できます。ただし、モニタモードになっ 信用できません。装走したときにどこの 番地を通ってきたかわからないからで す。もし、運がよければデータは大丈夫

RSTで再開始番地を操作

かもしれません。

次のように使われています。 答 RST 命令は、PC-8801シリーズでは

RST命令の使用

命	令	アドレス	項目名	機 能
RST	00H	0000	リセット	
RST	H80	0008	シンタックスチェック	RST(or CALL)の後に置かれているキャラクタ と(HL)とを比較して、一致しなかったら Syntax error、一致したらRST 10Hをして戻る
		0010	テキストからの 1文字読み込み	テキストから 1 文字または数を読み込む
RST	18H	0018	1 文字出力	CRTまたはLPTに1文字出力をする。A レジスタに文字コードを入れてRST 18Hを行う (E64C) = 0 CRTに出力 + 0 LPTに出力
RST :	20H	0020	HL、DE の比較	HL-DE(無符号)を行いフラグをセットする。Aレジスタはこわれる
RST	28H	0028	SGN (FAC)	FAC(単精度, 倍細度)の符号を調べる ー: A レジスタ=FF, NZ, M ロ: A レジスタ=0, Z, P +: A レジスタ=1, NZ, P
RST	30H	0 0 3 0	FACの型チェック	FACの型を調べる INT: Aレジスタ=FF, C, NZ, M, PE STR: Aレジスタ=00, C, Z, P, PE SNG: Aレジスタ=01, C, NZ, P, PO DBL: Aレジスタ=05, NC, NZ, P, PE
RST	38H	0038	ユーザ用 RST	モニタではプレイク・ポイント用またはコ マンド待ちに戻るときに使う

à la carte 7

DMAを 止めてしまうと・・・

グラフィックスの場合は、DMA処理を止めると、何かと便利だと聞くけど

データ転送が速く できる

DMA (Direct Memory Access) とは ダイレクト・メモリ・アクセスの略で、 通常使用されているCPU を使用しない でメモリとメモリ、メモリと I/O の間で 随接データのやりとりを行う方法です。 DMA 処理を行うためには、それ専用の ハードウェアが必要となりますが、データの転送がCPU を使用するより遠くで きるのが接承である。

グラフィックスでは DMAはいらない

PC-8801シリー までは DMA は Fに CRTC との間に優 用されています。 CRTC は定期的に VRAM のデータを 遊み込むのに優用されており DMA 処 埋を行っている脳は CPU は STOP し てしまいますので、CPU の見かけの処理 減度が遅くなります。

この DMA はテキストモードのとき だけ必要です。グラフィックス VRAM だけ使用しているのであれば DMA 処 卵は必要ありません。

そこで、DMA を止めてしまうと、30% ぐらいスピードアップできます。 テキス トモードで使用してもかまいませんが、 表示が消えてしまいます。ですから単な る計算のみを速く処理したいのであれば 非常に有効な方法です。

DMAストップで スピードをアップする

マシン語のプログラムでは P.253のような命令を入れておくと DMA のスト っプとスタートが簡単にできます。本書 のプログラムにこのプログラムを追加す るとスピードが速くなるのがわかります。

BASICプログラムで 使うには

また、BASIC プログラムでも使用できます。ただし、テキスト文字は削えますので DMA をスタートさせてからテキスト文字を表示させるようにすれば楽にできます。

● DMA のストップ

OUT81, 0または OUT 104, 0

WIDTH80, 25

このプログラムで実行にかかる時間は 約30秒です。しかし、20行を消して実行 すると約45秒かかります。15秒間は無駄 な時間を消費していることがわかりま す。プログラムを正大して、どんどん使 いましょう

●DMAのストップ&スタート

```
ORG
      START
      (SPSAVE), SP
    SP.STACK
      BC.5019H
      A.19H
                    テキスト画面の設定
      (BE684H), A
      A. BOH
      (@EGB8H).A
LD
      A. ØFFH
CALL
L.D
      A.00H
                   DMAをストップする
                   必要なプログラム
      BC,5019H
      (@EGB2H1.A
      (ME6B3H), A
                   DMAをスタートさせる
      A. ØE8H
                   くこれを忘れるとモニタに戻った
                     とき画面に何も表示されない)
      A.HOIL
      (0E6B9H),A
CALL
      COLOR
      SP, (SPSAVE)
     MON
              モニクに戻る
JP
```

END BASICでの例

```
18 REM ****** DMA STOP DEMO PROGRAM ******
28 OUT 81.8 TO 5888
9 FOR NA TO 5888
18 B-14
88 NEXT N
78 WIDTH 88.25
89 PRINT B- -18
```

à la carte 8

タンスのような バンク切り替え

わかったようでよくわからないのが、バン ク切り替え 単純明快に説明すると…

64Kバイトでは 足りないメモリ

バンク切り替えというのは簡単に言えば メモリの容量を増やす方法のひとつです。 PC-8801mkIISRに使用されているCPII は8ビット·CPUの Z-80A ですから、こ れで一度にメモリをアクセスできるのは FFFFH 帯地までの64K バイトです。

一告前に比べると64K バイトは速し て小さくないのですが、BASIC インタブ リタのプログラムの k きさが39K バイ ト以上もあると、テキストエリアが小さ くなってしまいます

また、グラフィックスの VRAM など は、48K バイトあり、とても64K バイト では不足してしまいます。

OUT. IN命会で メモリを増やす

そこで、メモリの大きさを増やすため に「OUT、IN」命令をうまく利用してい ます。「OUT, IN」命令は実行するには メインメモリと直接関係ないデバイス番 号 (00H-FFH まで) を指定する必要が あります。つまり、デバイス番号の I/O 番地にメモリをつけておくと、メインメ モリとデバイス番号につけたメモリを交 換することが可能です。この方法を利用 はパンクメモリということになります。

す。しかし、一度にアクセスできるメモ リの大きさは、64K バイトしかありませ

メモリをタンスにたとえて説明するレ P.255のようになります

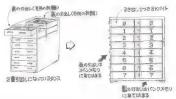
バンクメモリは タンス

二重引出しのタンスを例にとって考え ましょう。今は夏の季節としましょう。 すると冬川の衣類はほとんど着ないの で、奥のタンスの引出しにしまっておき

善段は表の引出しを利用していれば即に 合います。しかし、非常に冷え込んで冬 用のトレーナーなどを引き出したい場合 があります。しかし、冬田の引出しけ虫 の引出しにしまってあるので、すぐにト レーナーを出すことはできません。

そこで、一度表の引出しを引き抜いて 裏の引出しを引き出します。すると、表 の引出しはタンス上にはなくなります その代りに裏の引出しが表の引出しの位 置に来ます。トレーナーを出すと、裏の 引出しは奥においやられて、空いた位置 に再び表の引出しか入ります つまり. 表の引出しはメインメモリ、盟の引出し

*バンクメモリをタンスにたとえる



表の引出しと裏の引出しを入れ替えることをバンク切り替えといいます。この例は は二重の例ですが、実際には何重にもな ります。このバンク切り替えをする命令が「OUT、IN」命令なのです。

à la carte 9

バンク切り替えに似た ウィンドウ機能

テキストRAMの内容を読み出すときなどに使うウィンドウ機能とは何だろう。

ROMと重なったTEXTエリアは そのままでは読み出せない

ウィンドウ機能は N₈₈-BASIC モード (モード 1) での特殊な機能ですが、一 種のバンク切り替えです。バンクの大き さが1K バイト単位というだけです。

N₈₁-BASIC モードでは、CPU か0000 H 番地から7FFFH 番地までのメモリを 読み出すときは ROM から読み出し、 書 き込むときは、0000H 番地から7FFFH 番地までのテキスト RAM に書き込み ます。すると、テキストRAM の内容を 読み出すことができないので、テキスト RAM の内容を ROM と関係とい800日 驀進から83FH 豪地 に一日終してから 読み出します。ただし、これはソフトウ よアでサポートしているのではなく、ハ ードウェアで1動的に8000日 寿地から に書き込んでくれるのです。ですから、 この800日 驀地から83FFH 番地に書き 込めば、自動的にテキストRAM にも書 き込んでくれます。

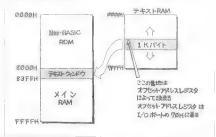
オフセットアドレスレジスタで 任意に読み書き

しかし、これでは1K バイトしか読み 再きができないので、これを0000H 番地 から7FFFH 番地のどこででもできるよ

*ウィンドウのメモリマップ

うに I/O ボートにオフセットアドレス レジスタをもっています。

このオフセットアドレスレジス # (上 使 7 ビットレジスタ)で任意の上位バイ トのアドレスをセットすることによっ て、自由に読み書きができます



オフセットアドレスレジスタは I/O
ボートの70日にあります。 この70日にア ドレスの | 校パイトをセットすれば良い のです。 オフセットアドレスレジスタに 関係する命令には次の3つからります。 ①オフセットアドレスレジスタに読み出 したいアドレスの上位パイトをセット OUT (70%), 上位パイト

②オフセットアドレスレジスタにセット されている上位パイトを読み出す ■ (70H)

③オフセットアドレスレジスタに I を加

える (出力データは何でもよい)

OUT (78H), ダミーデータ このウィンドウ機能をテストするには モニタコマンドの D, I, O コマンドを使 用すれば簡単にできます

1000H番地から1バイトを ウィンドウに表示

もし、1000日 番地から1K バイトをウィンドウに表示させるのであれば、

h] 070, 10回 とすれば、ウィンドウには1900日 番地か ら13FFH 番地まで表示されているはず です Dコマンドで確めてください また +100H 番地からならば、

[h 0 78, 00, 2

- このデータは何でもよい とすれば、オフセットアドレスレジスタ に1 が加えられて11H となり, 1100H 器 物から14FFH 番地主で表示されること

表示されている メモリアドレスを知りたい

ウィンドウに表示されているメモリ番 触を知りたいときは、次のようにします h] 1 70

23

23H というデータが減ってきたたら ば、2300H 番地から26FFH までという ことになります。つまり、返ってきたデ 00H をつけてそこから1K バイト分とい うことになります。 表示しているアドレ スは、

一つけ加える - Kbite分

2300H + 3FFH = 26FFH

ウィンドウに表示 されている実際の 示されている実 番地の始まり

ウィンドウに画 標の番曲の終り

とすれば簡単に求めることができます。

どうして面倒な 窓を作ったのか

アミして、このような理解しにくい機 能をとり入れたのでしょうか、その理由 は、BASIC 命令にグラフィックス命令な どを追加したため、BASIC のインタブリ ムです (約)32K バイト、このもとになっ たN-BASICインタブリタは24Kパイ

そのためにテキスト RAM が不足し たので、バンク切り替えでテキスト RAM を助やさむければならなかったの です。そこで単なるバンク切り替えでテ キスト RAM を増やすと、BASIC イン タプリタは頻繁にテキスト RAM をア クセスするので、実行速度が低下します。 インメモリに持ち、できるだけバンク切 り替えをしなくてすむようにしたわけで

つまり、IK バイト以内であるならば バンク切り替えを必要としないのです。 1K バイトより大きくても、オフセット スをセットするだけなので、1回のバン ク切り持えで済みます (弊派にバンク団 り替えをすると、もとのメインメモリに 戻るためにはもう1回必要です)

貫上の理由からウィンドウ機能という 方法を採用したのだと思われます。

à la carte 10

メモリモードを切り替えるには

メモリモードを切り替えれば、ウィンドウを使わずに直接RAMに書き込める。

PC-8801mkHSR は3つのメモリモー ドを持っています。

₹-F0 N-BASIC ₹-F

€- F 1 N₈₄-BASIC €- F €- F 2 64K RAM €- F

以上の3つのモードをプログラムで切り替えることができます。一番多く使用されるモード1とモード2の切り替えについて説明しましょう。

どんなときに モード切り替えが必要か

モード切り替えは次のようなときに必 要となります。

①64K RAM モードでプログラムを動か す場合

64K RAM モードで使用する場合は 最初にメモリモードを切り替えて使用す る場合が多いようです

このモードで多く使用されているの は、他の DOS (たとえば、CP/Mや UCSD)を使用する場合です。

②テキスト RAM をウィンドウを通さず にアクセスしたい場合

ウィンドウは1K バイトなので多量の テキストエリアのデータをアクセスでき ません。そこで、すばやく処理するため 、時的に RAM モードにしてからテ ータを転送したい場合などに使用されま す。ゲームソフトに多いようです

③84K RAM モードで動いているプログ ラムから、Nas-BASIC の ROM 内ルーチ ンを呼びたい場合

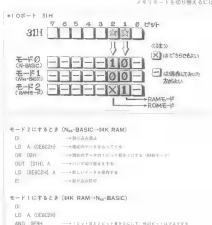
計算に ROM 内ルーチンを使用して、 プログラムを簡単にしたい場合などに使 われます。この場合、BASIC のワークエ リアをそのまま持っていなければなりま せん。

モードを 切り替えるには

メモリモードをコントロールしている のは、I/O ボートの3IHの1ビット目と 2ビット目です。このビットをコントロ ~ルすればいいのです

モード変更するには、2 ビットだけで ないのですが、他のビットは他の用絵に 使用されているために、勝手に変えるこ とはできません、そこで、[OR、AND] 命令を使用します、現在の I/O ボートの 31日のデータは EGC2H (Nas-BASIC の ド場)に保存されていますので、この データを利用すればいいことになりま す、モード1、2 にするには P. 259のよう にします。

バンク切り替えを行うときは、必ず割 り込みを禁止しなければなりません。64 K RAM 実行中は割り込みを禁止しな



いと、Nes-BASIC ROM のルーチンを 使用する関係で発走をおこします。

64K RAMを使った サンプルプログラム

OUT (31H), A LD (0E6C2H)

順理がわかったところで簡単をサンプ ルプログラムを作ってみましょう、Noo-BASIC モードでは、ウィンドウを使用す ることによって、0000H から始まるメモ りにデータを書き込むことが簡単にでき ます.

つまり、モニタモードでSコマンドな どを使用して、0000H 番地に書き込め ば、自動的に RAM 側に滲ぎ込まれるの です、したがって、0000日 番地に簡単な プログラムをサブルーチンとして恵き込

マシン語アラカルト

みます。

64K RAM モードにして、サブルーチ ンをコールしてみましょう。64K RAM モードにした場合はモニタも消えてしま うので、何らかのプログラムを入れてお かないと暴走してしまうからです。

プログラムは次のようにします。

●メインプログラム

· Nou-BASIC E - F (E - F1) T-B4K RAM

モード(モード2)に切り替える

・0000H 番地からのサブルーチンをコー ルする

・64K RAM モード (モード2) から Nas-BASICモードに切り替える

モニタに戻る ●サブルーチン

*DOIDH 驀地から、順次、AAH というデー タを FFH 同書き込む:

			/ *Uni (E-F1)nt	

B988	START:	EQU	8B966H	
		ORG	START	
8988 F3 8981 3AC2E6 8984 F682 8986 D331 8988 32C2E6	:	OR	A.(0E6C2H) 02H (31H).A (0E6C2H).A	: 77つミ キンシ : MEMORY MODE
BBBB CDBBBB		CALL	69869	
B98E 3AC2E6 B911 E6F9 B913 D331 B915 32C2E6 B918 FB		AND	A.(8E6C2H) 8F9H (31H).A (8E6C2H).A	
8919 FF		RST	38H	:モニターニ モトニル

●ウィンドウに表れているプログラムを モニタを使って逆アセンブルした

END

8888	21	0010	LX1	H.0010
8003	3E	AA	IVM	A.AA
8885	9.6	FF	MVI	B.FF
8007	77		MOV	M.A
8888	23		INX	H
8009	95		DCR	В
800A	C2	9997	JNZ	8987
SOUD	C9		RET	

ウィンドウを使用して、メインプログ ラムを実行させる前に、0000H 番地から 書き込まなくてはなりません。ウィンド ウの関係で番地が8000H番地からにな っていますが、実際には0000H 番地から 書き込まれています。 サブルーチンはイ ンテルニーモニックになっていますが, これは、モニタのLコマンドで遊アセン

B91A

これを書き込んだ後にプログラムを実 行すると、すぐにモニタに戻ります そ こで、Eコマンドで8000H 番地からのウ (ンドウを見てください、しっかりと宝 行されていることがわかります。ここで はメインプログラムを説明するために簡 単にし、サブルーチンをメインプログラ ム中に入れていませんが、応用する場合 には CALL 命令を実行する前に、サブル

- チンの転送プログラムを入れておく必 要があります

(MOOH 番地のサブルーチンをメイン プログラムに入れたプログラムは次のよ うになります。これで、ウィンドウを値 川せず RAM 全体に書き込み、読み込み ができるようになりましたね、これを少 し応用すると G-VRAM をデータ保存 の RAM にしたりすることが可能です。

		****	******	****
	メモリ	£-1"	ノ キリカエ (モート~)カラ	2)
	PROG	RAM N	AME> MB1.e	(WRITE & SUBG®)
		30 48 48 46 46	*******	经收款股票的证据
none.	START:	EQU	8B988H	
	-	ORG	START	
F3 B901 3AC2E6 B904 F602 B906 D331 B906 32C2E6		DI LD OR OUT LD	A.(0E6C2H) 62H (31H).A (0E6C2H).A	:7932 #55 : MEMORY MODE 2
B908 212589 B90E 118000 B911 010F00 B914 ED80		LD LD LD LDIR	HL.SSUB DE.0000H BC.000FH	:8688H = 9716-F2 9 5434
B916 CD6888		CALL	8666H	
B919 3AC2E6 B91C E6F9 B91E D331 B920 32C2E6		AND	A.(8E6C2H) 8F9H (31H).A (8E6C2H).A	: MEMORY MODE !
8923 FB 8924 FF		E1 RST	38H	:モニターニ モトトル
8925 211886 8928 3EAA 892A 86FF 892C 77 892D 23 892E 85 892F C26788 8932 C9 8933 8888	SSUB:	DB DB DB DB DB DB	21H.18H.98H 3EH.8AAH 86H.8FFH 77H 23H 95H 6C2H.87H.88H 6C9H 88H.88H	:LD HL.0819H :LD A.AAH :LD B.FFH :LOOP:LD (HL).A :INC HL :DEC B :JP NZ.LOOP
B935		END		

●サブルーチンと実行無事

	サフルーチン 一																
8899																	1 Dr W 5 /
8818	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	ILLIXIZATION IN THE STREET
8826											AA						
8838	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	
8848	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	IIIIIIIIIIIIIIII
8858	AA.	AA.	AA	AA	AA	*************											
8060	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	INTERTEGRATION IN
8870	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	IIIIIIIIIIIIIIIIIII
8888	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	THIFTHWARKERS
8899	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	88	AA	AA	SA	AA	IIITITITITITITITI
BAGB	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	
8888	AA	AA	AA	AA	AA	AA	AA	AA	88	AA	AA	AA	AA	AA	AA	AA	RICELLICISISSE
89C8	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
8608	AA	AA	AA	AA	88	AA	AA	AA	TITITITITITITITI								
BBEB	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	TITITITITITITITITITITITITITITITITITITI
80F0	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	NAMED AND ADDRESS OF THE PERSON OF THE PERSO

注:ウィンドウを使用しているので、8000H書地からになって いるが、実際は60000H書地 実行結集

à la carte 11

NEWしたプログラムを 復活するには

BASIC プログラムを間違えて消してしまったとき、何とか復活できないか?

間濾ってNEW 命令を実行したり、 | STOP| キーを押きないで、(RESET) キーを押してはまうと、BASIC のプログラ ムが消えてしまいます。こんなときにブ ログラムを視済する方法はないのでしょ うか、マシン語を利用すれば復済が可能 です。

リンクポインタのみを 復活すればよい

NEW 命令や RESET をかけるとメモ リにすべて00H が書き込まれるわけで なく、リンクポインタといわれる部分と、 終りをボすポインタのみが消されるだけ なのです。プログラムを復活するにはリ ンクポインタのみを復活すればよいので す

DISK-BASIC についてくる「dskut 1,n88」のプログラムを例にとって説明 してみましょう、NEW 命令で消される のはテキストエリアの最初のリンクポイ ンタが90日になるだけですから、最初の リクガインタを見つけて、正しいデー タをセットするだけです。

一般的に初めの1行はP.263のような 構成をしています。

*リンクポインタの1行目

10行 0001 0700 0A00 8F 00
 「格納アドレス セリンクオセンタ ®行番号 4 RAMの 8 F行の終りの目印

①終歴アドレス

新しいプログラムはすべて、テキスト エリアの0001H 番地から格納されます。 (プリンクポインタ

次の行が格納されている番地、2-80 CPUの関係に上に位べイトと下位バイト が入れ替わっています。このリンクボイ ンタのデータが0000日の場合は、特別な 意味があります。つまり、

意味があります。つまり、 0000H…BASIC プログラムの終り を指し示します。したがって0000H以外 のデータならば、次の行があるというこ とになります

(3)行番号

BASIC で使用されている行為号、16.進 数の4桁なので、BASIC で使用できる行 寄号は抽高で65535行です。この例では0 A00H の上位・下位バイトを入れ替える と000AH なので BASIC の行番号は10 行

(4) BASIC の中間言題

BASIC コマンドはそのままストア(格納) されるのではなく、1パイトから2 パイトの中間、清部に変換されてからスト できれます。中間、清部がわかれば、マシン語でBASIC のプログラムの書き替え や解析も可能となります。

⑤行の融りの目印

この00H は行の終りを示すデータで す、ここの00H は次のリンクポインタを 見つける際の唯一の手がかりとなってい ます。

プログラムの 復活

「dskutl, n88」のプログラムを使用してみましょう。実際にメモリがどのよう にストアされているか、ウィンドウを通 してテキストエリアの0000H 番地を表 示させると、次のようになります

●テキストエリアの0000H番地

hidsege.seff 8880 98 87 80 E8 83 8F 80 40 60 F2 83 BF 28 28 28 28 8818 28 44 49 53 4B 28 55 54 49 4C 49 54 59 28 66 6F DISK UTILITY fo 8020 72 20 50 43 2D 36 38 30 31 4D 4B 32 20 28 28 31 r PC-8881MK2 (| 803E FE 64 72 69 76 65 28 73 74 65 6D 28 29 88 drive system i 8848 46 68 93 BF 88 88 86 94 8F 28 28 28 28 43 8858 GF 78 31 39 38 opyright (C) 198 8868 33 88 79 28 0.5 SF SE 61 60 29 43 SE 8878 8 78 75 28 44 69 76 28 4E 45 43 88 mputer Div. NEC 8888 86 MI 18 84 8F gg Bi 88 1A 64 8F 28 28 28 28 28 8098 100 28 26 28 28 28 28 28 56 65 69 6F 6E BBAB 28 31 2E 38 28 28 28 38 33 OF 38 39 2F 32 32 29 1.8 (83/89/22) 88B8 88 B7 24 84 BE 28 FA BU 28 28 92 28 20 0 5 4 0 26 34 8F 28 63 68 6B 78 & :+0 check Ex 88D8 78 61 6F 53 74 61 74 65 6D 55 6F panded Statement 28 28 88E8 28 58 61 63 68 61 67 65 2F 88 F7 88 38 84 RHEB 92 28 2C BC FF CF BB 83 81 42 84 28 28 AB 28 41

マシン語アラカルト

この中で大切なのは1行目です。この います。その秘密をこれから説明してい 行にプログラムの復活の秘密が隠されて きましょう。

* ウィンドウの1行目



初めのリンクポインタ0700H に注目 命令で消してしまうと1行目はどうなる してください。このプログラムを NEW でしょうか。



初めのリンクボインタに0000H が書き込まれる以外は、すべてメモリ上に残っています。ここに次のリンクボインタのデータを書き込めば、プログラムは復活するはずです。

復活の手がかりは 00H

リンクポインタの復活のためには次の リンクポインタを探さなければなりませ ん、それには行の終りの日印である00H を探せばよいことになります。ただし、 (場話させようとするリンクボンタの次 の2×イトは行番号ですから、どちらか に00H が入っていることもありますの で、4×イト日以降の00H を探します。 00日を探していくと、8000H 美地に00H があります。すると次の番地からリンク ボインクが始まります。次の番地が69ンク 用番地ですが、現活しようとするリンク ボインタに0780H と書き込んではいけ ません それはウィンドウを利用してい るので8000H 番地代になっていますが、

本当のメモリは0000日帯地代から始ま っているからです。したがって"0700H" と書き込まなくてはいけません。

*復活したリンクポインタ



これでリンクポインタが復活!まし た BASIC モードに戻って LIST をとっ てください、BASIC プログラムが復活し てるはずです

初めのリンクポインタに0000Hが書 ③ h] e8000[]…ウィンドウをエディッ かれていると、NEW 命令を実行したと きと同じになります。また違う位置のリ ンクボインタのところに0000H が事か れているとプログラムの終りを意味しま す、したがって、プログラムの終りを探 すには00Hが3バイト書き込まれてい る器地を探せば上いのです

BASICプログラム 復活の手順

8000日 帯地からのデータは、0000日 & 地から始まるテキストエリアとは関り主 せん、そこで、どうしてもウィンドウが 表示している番地を0000H番地にした ければなりません。ウィンドウを0000H 巻地にするには.

h] 070, 0 [

ウィンドウの表示書館を8089H 善娘からにする この命令を実行することによって、ウィ ンドウが0000日 番地になります

復活の手順

① MON [] …モニタモードにする ② h] 070, 0 ... ウィンドウを0000H 器 地にする

トモードにする

④次のリンクポインタの位置を探す (OOH を手がかりにする。OOH の次の位

置が次のリンクポインタ) ⑤上位バイトと下位バイトを入れ替えて

リンクポインタに書き込む ⑥エディットモードを終了して、BASIC モードに戻る

⑦確認のため、LISTをとる (8)38 i)

簡単なプログラムで 復活の

プログラムの復活の方法は大事なこと なので、簡単なプログラムを使って練習 してみましょう。ます"NEW 「"でプロ グラムを消します。

モニタでダンプリストをとってみます (P.266泰州)

●消すBASICプログラム

10 REM **** TEST PROGRAM **** 26 PRINT " *** RETEST PROGRAM ****" 30 END

●ダンプリスト

```
8868 88 IF 88 8A 88 8F 28 2A 2A
                                2A 2A 2A 28 54 45 53
8818 54 20 58 52 4F 47 52 41 4D 28 2A 2A 2A 2A 88 48 T PROGRAM **** #
8020 00 14 00 91
                 28
                      28
                                2A 20 52 45 54 45 53
8838 54 28 58 52 4F 47 52 41 4D
                                   2A 2A 2A 2A 22 88 T PRO
8F 28 28 28 28 43 F
                                28
                                                      T PROGRAM ****
8846 46 86 IE 68 81 69 86 88
                                84 8F 28 28
8858 6F 78 79 72 69 67 68 74 28 28 43 29 28 31 39 38 opyright (C) 195
8868 33 28 62 79
                 28 58 65
                                GF SE 61 SC 20 43 SF II by Personal Co
8070 SD 78 75
                    72 20
                          44 69 76
                                       28 4E 45 43 88 mputer Div. NEC
8080 86 B0 10 04 8F 00 B1
                          88
                             1A 84 SF
                                          28 28 28 28 8 + 7 +
8898
    28
              28
                 28
                    28
                       28
                             28
                                   65
    28
        31
              38 28
                    28 28
                                    38 39
                                                        1.8 (83/89/22)
8888 88 87 88 24 84 8F 88 EA 88
                                2E 84 28
                                            92 28
                                                       + 5 + 0 .
SECH BC FF
          E5 28 3A 8F E9
                          28
                             63 68
                                   65 63 6B 2B 45 78
                                                       b :+♥ check Ex
8000 70 51 6E 64 65 64 20 53 74 61
                                   74 65 6D 65 6E 74
                                                      panded Statement
88E8 28 58 61 63 6B 61 67 65 2E 88 F7 88 38 84 28 28
                                                       Package, 8 8
88F8 92 28 2C 8C FF CF 88 83 81 42 84 28 28 AB 28 41 4 , 7 B
```

次の問題を解いてみましょう。
ります。

間 I 初めのリンクポインタが0000H と

リンクポインタの位置を探してくだ

(ヒント) 初めのリンクポインタの位置 から4バイト以降の00Hを探す

間2 このプログラムを復活させてくだ

間3 このプログラムは何番地で終って いるでしょうか、

プログラム復活の

RR 1

00H のある位置は801FH 条地なので、 次の器域801FH 器域となります。したが って、メモリ上の番地は001FH 番地とな

B 2

なっています。ここに書き込む次の 次のリンクポインタの位置が001FH

8000 00 00 00 0A 00

1F 00

8001番地に1FH を書き込むだけでよ

BASICモードに戻ればリストがとれ

ます. 間 3

00H が3 個連続している位置を探す か、最後のリンクポインタのデータをみ ればよいのです。答は0046Hとなりま す、

à la carte 12

自由にファンクションキーを設定したい

BASICのKEY命令のように、マシン語でも 自由に設定できないか。

マシン語でも、BASICの「KEY」命令 のように自由にファンクションキーを設 定できたら、何かと便利です。比較的樂 にしようと思ったら、ROM 内ルーチン を呼びます

■ ファンクションキー・データの あるアドレス

ファンクションキーのデータは次のア といるにもります

FVX	こめります。	
F. 1	E6F2H~E701H	16/17 1
F. 2	E702H~E711H	16バイト
F. 3	E712H~E721H	16/17 1
F. 4	E722H~E731H	16/11/
F, 5	E732H~E741H	16バイト
F. 6	E742H~E751H	16バイト
F. 7	E752H~E761H	16/17
F. 8	E762H~E771H	16/17
F. 9	E772H~E781H	16バイト
F. 10	E782H~E791H	16/17

各キーはそれぞれ店がイトということ になっていますが、次のファンクション キーのデークと区別するために100日が 必要なので、実際に当業できるま字は45 文字までになります。CRTトに出る1つ のファンクションキーの内容長がは、11 文字ですから光がな大きさでしょう。も し表示される文字が11文字でもよいから もうかし良くしたい場合は、文字例の最 後に00H を入れなければ次のファンクションキーのデータとつながります。この場合は、31文字として扱われます。

ファンクションキーのデータ を変える手順

①ファンクションキーのデータエリアに ASCII コードで任意の文字を書く(モニタ を使ってもよい)

② E6B8H に FFH を書き込む

③ ROM 内ルーチンの「3F7AH」をコール ④終了

実際に実行してみましょう。モニタの Dコマンドは、18バイト以上はスター ト・アドレスとエンド・アドレスを指定 しなければならないので、再週かかかり ます。そこで、いつら後まっている萎地 をファンタションキーに設定しておけてい る番地を設定してみましょう。マシン胡 のスタート・アドレスは B900H 番地で す。

ファンクションキーに次のように割り

当てます。

F1 → 08900, BFFF F2 → L8900, BFFF

F3----- A8900

F4---- MB900, BFFF, D900

F5 → GB900 %

マシン語アラカルト

これらのデータをブロック転送命令の プログラムは次のようになります。 「LDIR」命令でデータ転送しています。

■ファンクションキーのコントロール

			***	****	计划设计设置操作情報程序公司公司	
		7720	ションキー	ノ コントロール		
		PROG	RAM N	AME>FC1.e		
			****	******	***********	
B908 BFFF 3F79 E6F2 E702 E712 E722 E732		START: DEND: FCSUB: FK1: FK2: FK3: FK4: FK4: FK5:	EQU	88988H 88FFFH 3F79H 8E6F2H 8E782H 8E712H 8E722H 8E732H		
			ORG	START		
	3EFF 32B8E6		LD LD	A.BFFH (BE6B8H).A	:F.KEY ON	
8988 8988	2148B9 11F2E6 818A88 EDB®		LD LD LD LDIR	HL.FCHRI DE.FKI BC.800AH		
8913 8916	214AB9 1182E7 616A88 EDB8		LD LD LD LDIR	HL.FCHR2 DE.FK2 BC.068AH		
B91E B921	215489 1112E7 010500 EDB0		LD LD LD LD I R	HL.FCHR3 DE.FK3 BC.8885H		
B929 B92C	215989 1122E7 018F08 EDB0		LD LD LD LD IR	RL.FCHR4 DE.FK4 BC.008FH		
8934 8937	216889 1132E7 818688 EDB0		LD LD LD LD IR	HL.FCHR5 DE.FK5 BC.8886H		
393C	CD793F		CALL	FCSUB		
893F	FF		RST	38H		
3944	44423938 382C4246 4646	FCHR1:	DB	.DB386'BEEL.		
394A 394E	4C423930 392C4246 4646	FCHR2:	DB	'LB900.BFFF'		
	41423938	FCHR3:	100	'AB988'		

8959 4D423938 FCHR4: IIII 'MB988, BFFF, D988' B95D 382C4246 B961 46462C44 B965 393030 8968 47423930 FCHR5: DE 'GB988',8DH B96C 300D B96E

●F220H器納からのダンプリスト

```
F228 3E FF 32 B8 E6 21 68 F2 11
F238 21 6A F2 11 82 E7 81 8A 88 ED B8 21 74 F2 11 12
                                                     1.15 F 0-1tm
F240 E7 01 05 08 ED B0 21 79 F2 11
                                  22 E7 81 8F 88 ED
                                                         O-TYE "F C
F258 B8 21 88 F2 11 32 E7 81 86 88 ED 80 CD 79 3F FF
                                                     -11 E 2F
F268 44 42 39 38 38 2C
                      42 46 46 46 4C 42 39 38 38 2C
                                                     DB900 BEFFLB900
F278 42 46 46 46 41 42
                      39 MM 38 4D 42 39
                                         38 38 20
                                                      BFFFAB900MB900.B
F288 46 46 46 2C 44 39 38 III 47 42 39 38 38 8D FF 88
                                                     FFF. D986G8988
```

しかし、このプログラムでは、他のブ ログラムを走らせることができません。 他のプログラムを走らせるためにはこの ままのアドレスでは長合が悪いので、S コマンドを使用して、F220H 番地から16 進数で入力します。 その後で "GF220 [7]"

と入力しますとモニタに成ります Non-BASICのワークエリアを壊していない ので、どのような命令でも使用できます

à la carte 13

VRAM* 移動させる

SRのV1モードでは、F3C8H番地から始まっ ているVRAMを動かせる。

N-BASICではテキスト VRAM は F300日 番地からと決まっていましたが、 N.a.-BASICでは自由に設定ができま す、PC-8801シリーズでは、初期設定が ているのは次の番地です。 F3C8H 番地からになっているにすぎま E6C4H →下位アドレス せん、したがって、Nan-BASICのワーク エリアを助さない範囲なら VRAM を自 由に設定できますが、コールド・リセッ

トなどがかかるともとの F3C8H 番地に 灰ってしまいます.

VRAMのスタート・アドレスを決め

(初期設定 CRH)

E6C5H →上位アドレス (初期設定 F3H)

CRTコントローラを 初期設定すればよい

このドレスをSコロンドをとで適当 に書き替えると、入力した文字は表示さ れません。これはCRTコントローラ (DMAを含むしが書き替えたアドレス を正しいドトレスとして参照しているか らです。すなわち、CRTコントローラが 切削減定されていないので表示できない ででると、完全にVRAMエリアは築化 します、幼馴波定の一部の側値モードの ときに使用したものと金(同じもので

す。ただし、初期設定してしまうと、今 までの立つけるベア治えます

VRAMスタート番地を D000H番地からに

VRAMのエリアを変化させて、スタ ト・アドレスを D000H 番地からにす るプログラムを作ってみましょう。下順 は次のようにします。 ()画面を タリアする ② E604H と F605H 番地に新しいスター ト・アドレスをセットする ③3時期設定のルーチンを呼ぶ これでOKです、プログラムを示します。

●テキストVRAMの移動

	-	* HE III CO.	COLOR TE	XT MODE ***	***
	TEXT	VRAM .	/ 410 (F3C8H-	->D8@BH1	
	PROGRA	M NA	ME> MVRAM.	e	
			有安全的 教育的公司	沙沙城州党县司安县县山	******
1988 1F05 1626 1F6A	STRAT: CLS: MON: COLOR:	EQU	88988H 5F8EH 8E626H 6F6AH		
		ORG	STRAT		
1908 CD8E5F		CALL	CLS	CRT	CLEAR
983 3E08 1985 32C4E6 1988 3ED8 1988 32C5E6		LD LD LD	A.88H (BEGC4H),A A.8D8H (BEGCSH),A	:VRAM	ADDRESS>D000H
990 011950 1910 3E01 1912 3282E6 1915 3E19 1917 3283E6 1918 3E88 1910 3284E6 1917 3E08 1921 3288E6 1924 3EFF 1924 3EFF 1925 3289E6 1929 0D6A6F			BC.5819H A.81H (8E682H),A A.19H (8E683H),A A.8E8H (8E684H),A A.8E8H (4E686H),A A.8FFH (4E686H),A		

B92C C326E6

JP MON

B92F

END

VRAM のスタート・アドレスが D000H 番地になります。したがって、F3C8H 番 地以降にキャラクタ・コードをセットし ても、CRT には表示されません。Sコマ ンドを利用して D000H 番地をドにキャ い コードに対応したキャラクタが表示 ート・アドレスは D000H 番地となりま t toll C BASIC +- FC | This. Poke &HD000, &H41 .

れます。

実験後は必ず リセットをかける

実験が終ったたらば、必ずリセットを かけてください、リセットしないで、他 のプログラムを生行させると基本した り、正しく表示されません。必ず正しい VRAM のスタート・アドレスにかっト

VRAM のスタート・アドレスを自由 に設定できるということは、テキスト画 面を2両面持つことも可能なわけです。 CRTコントローラ (DMA も含む) を初 としてくがさい た上に「A」が表示さ 期設定しないようにすればできます。

à la carte 14

ユーザが使える 未使用命令

Nas-BASICには、どんな未使用命令があり、 どのように使えるか、

未使用命令は

N...-BASICでは、使用されていない 未使用命令が8個あります。それは次の キーワードです。

使用されていないキーワード

(I) WBYTE

(2) RBYTE

(3) POLL

(4) ISET

(5) IRESET

(7) IEEE (R) CMD

ただし、 (8の CMD は拡張命令の第1 キーワードとして健用されているので使 川できません 拡張命令を切り放してお

けば使用できます。

もともと、これらのキーワードは
IEEE インターフェースの制御用したり
用意されたものですが、Nar BASICではサポートしていないので、未使用命令
となっているのです。これらの命令を実
行しても「Feature not available」とい
うエラーメーッセージが接示されるだけ
で使用できません。しかし、これらのキーワードはジャンプテーブルを変えれば
ユーザが自由に使用できます。これは、

非常に便利な使い方です。たとえばPオ ブションの解除にはこの未使用命令を使

BASIC命令を使って 未使用命令を実行

用しないとできません

実際に独自の命令を作るとなると、結構時間を食われるので、BASIに命令を実 行するようにしてみます。たとえば未使 用命令の「POLL」を BASIに命令の 「PRINT」命令にするということです。 したがって、POLLを実行すれば PRINT命令と全く同じに使用できます。これをうまく利用するとプログラム

未使用命令を使用するためには、それ らのキーワードのジャンプテーブル番地 を知る必要があります。それらは次のよ うになっています。

キーワードのジャンプテーブル番地

7 / 1 ///	(- / / / mm
キーワード	テーブル書地
① WBYTE	EEAIH
② RBYTE	EEA4H
③ POLL	EEA7H
4 ISET	EEAAH
(5) IRESET	EEADH
⑤ STATUS	EEB0H

① STATUS	EEB3H
® CMD	EEB6H
(9) IEEE	EEB9H

⑥、⑦のSTATUSというキーワードは 同じものですが、ジャンプアドレスが2 つ用意されています、モニタを使用して、 選不とンプルしてみると次のようになり ます。

●ジャンプアドレスを逆アセンブル

					キーワート
EEA1	C3	4DC1	JMP	4DC1	WBYTE
EEA4	C3	4DC1	JMP	4DC1	RBYTE
EEA7	C3	4DC1	JMP	4DC1	POLL
EEAA	C3	4DC1	JMP	4DC1	ISET
EEAD	C3	4DC1	JMP	4DC1	IRESET
EEBB	C3	4DC1	JMP	4DC1	STATUS
EEB3	C3	4DC1	JMP	4DC1	STATUS
EEB6	C3	4DC1	JMP	4DC1	CMD
EEB9	C3	4DC1	JMP	4DC1	I EEE

これらのジャンプテーブルにはすべて [JP 4DC1H] が入っています。ですか らこれらのキーワードを実持するとすべ て、4DC1H 番地にジャンプします。4DC1 H 番地は [Feature not available] とい エラーメッセージを表示するルーチン です。したがって4DC1H 番地以外へジャンプすれば、これらの命令も使用でき もようになります。

未使用命令を次のような BASIC 命令 と対応させてみましょう

未使用命令の キーワード	BASICの キーワード	BASICのエ ントリポイント
WBYTE	CLS	7185H
RBYTE	INPUT	102DH
POLL	PRINT	0E54H
ISET	RET	009CH
IRESET	NEW	77DDH
STATUS	LIST	18D9H
CMD	RUN	0B7CH

■モニタを使って逆アセンブルしたもの

EEA1	C3	71B5	JMP	7185
EEA4	C3	182D	JMP	1020
EEA7	C3	8E54	JMP	8E54
EEAA	C3	ØC9C	JMP	ØC90
EEAD	C3	77DD	JMP	77DE
EEB0	C3	18D9	JMP	18D5
EEB3	C3	18D9	JMP	1809
EEB6	C3	087C	JMP	0B70
EEB9	C3	4DC1	JMP	4DC1

■新しいキーワードで作ったプログラム

18	REM was NEW	KEY	WORD	TEST	***
28	WBYTE				
38	RBYTE D				
48	A=DeD				
58	POLL "D*D=	":A:			

78 END

●BASICコマンドで書き直すと…

- 18 REM *** NEW KEY WORD TEST (BASIC) ***
- 30 INPUT D
- 50 PRINT "DWD= ";A
- 68 GOTO 38

BASICのエントリポイントを未練用 命令のシャンプテープルにSコマンドな と機用して書き込みます。 書き終ったな らば、BASICモードに戻って、新しいキ ーワードでプログラムを作って実行させ てみてください、しっかりと動くはずで す。

Fのプログラムを見てください。これ は、新しいキーワードで作ったプログラ ムです、BASIC に熱知している人であれ ほ見当がつくかも知れませんが、ちょっ とただけでは何のプログラムかわかり ません。

これを BASIC コマンドで書き直すと 上のようになります。

未使用命令のジャンプテーブル群はリ

セット時に必ず4DC1H が書き込まれま すので、書き込んだデータはディスクに 保存しておくと良いでしょう。

このジャンプテーブルをうまく利用すると、プログラムの実行キーワードを作って、他の人にプログラムを実行させないようにすることも可能です。自分でエー夫してみてください

(セント)

プログラムをロードした後にジャンプ テーブルを書き替えるプログラムを実行 させるか、BASIC プログラム上に、ある キーワードでしか実行できないようなプ ログラムエリアを作っておくと良いでしょう。

à la carte 15

プロテクト はずし

Pオプション付、すなわちプロテクトのか かったプログラムのリストをとる

モニタにさえ 入れない

Pオプション付のプログラムは、プロ グラムリストはもちろんチニタにコテも 入ることはできません Pオプションを つけると次のコマンドが宝行できたくた ります。

(6) BSAVE

(I) LIST

(2) LLIST (7) BLOAD (3) MON

(4) POKE (5) PEEK

外にマシン語を扱う命令までも実行でき ません。マシン語が実行できれば、簡単 にPオプションを解除できます。

解除の原理は

Pオプションの解除の原理は糖単で、 次に示す器據へ00H を書き込むだけで 良いのです。

ワークエリア

EC29H OOH----→Pオプションなし ONH 以外→Pオプションあり

(-- #9 to 20H)

したがって、EC29H 番地に00H を書き

込めばよいのですが、前に説明したよう に、MON、POKE 命令が使用できないの で、書き込む方法がありません。Pオブ ションを解除するにはどうしてもマシン 語のプログラムを実行させなければなり ません。一番良いのは、チニタチードに なってくれることなのですが……

キーワードは 未使用命令

モニタモードに入る方法を考えました う、もし、Pオプションがかかっていな ければ、「MON DIで入ることができま Pオプションをつけるとリスト命令以 す。しかし、MON 命令を使用せずにモニ 夕に入る方法を考えなければなりませ ん、いわゆる下約語以外の命令を実行さ せようとしても、「Syntax error」がでる だけです。したがって、BASIC の予約語 の中で、使用していない命令(未使用金 令)を使用してみようというわけです。 未使用命令には次のような予約語(キ

> →ワード)があります。 未使用の予約額(キーワード)とジャン

プテーブル

○ WBYTE FFAIH (2) RBYTE FEAAH (3) POLL EEA7H

(4) ISET EEAAH (5) IRESET EEADH

⑥ STATUS	EE80H
① STATUS	EEB3H
® CMD	EEB6H
@ IEEE	CERON

この中で、実用的に使用できるのは①

・②までです。これらのジャンプテープ

・麻藤はにはずな「 JP 4DCHI と きかれており、この4DCH 基地へジャンプテープルは、RAM 領域にあり、許さ替えが可能です。そこで、このジャンプテーブルの泉代を基地をマシン部のプログラムの先頭にしてあげれば、モニタモードにすることが可能です。そこで、この水は用をからするこのを使用してみまれた場所をありまつるを使用してみまれた。

しょう。 POLL ―モニタモードに入る ISET ― P オプションを解除して BASI これら2つの未使用命令の番地を書き 棒えて実行可能な命令にしますので、これらの命令を実行したときにエラーがで ません、なお、未使用命令はマシン語の プログラムに入るキーとして使用してい るだけです。

Pオプション解除

ISET 命令を使用してPオプションを 解除するプログラムを作りましょう。

BASIC では ISET 命令を実行すると、 EEAAH 番地に飛んできますので、ここ の帯地を次のように書き替えます。 EEAAH JP 4DC1H

JP OF22AH

0F22AH 番地からに EC29H 番地を00H にするプログラムを入れるだけです。 ア ログラムとダンプリストを示します。

Cモードに戻る ●Pオプション■除

		***	*** P OPTION FREE PROGRAM ********
	P #7") 92 J	n2°2
	PROGRA	AM NA	ME> POF.e
	*****	N 80 NG NG NG	********************************
F228 50E5	START: REBAS:		8F228H 5 8E5H ← BASIC〜戻る
		ORG	START
228 3E2A 222 32ABEE 225 3EF2 227 32ACEE 22A 3E88 22C 3229EC 22F C3E558	•	LD	A、2AH (BECABH)、A A、8PCH (BECACH)、A (BECACH)、A (BEC2SH)、A REBAS
7232		END	

●ダンプリスト

F228 3E 2A 32 AB EE 3E F2 32 AC EE 3E 88 32 29 EC C3 F238 BB 58 68

非常に短いプログラムですから、16進数 で直接 F220日 番地から入力した方がよ いでしょう

このプログラムを 保存するには

プログラムの保存は次のようにしてく ださい、BASIC モードにしてから、 BSAVE "pof", &HF220, &H2FH 🕣

これで保存できます。このプログラム を実行する場合は、 BLOAD "pof", &HF220, r ☑

プロテクトはずしの 手順

このプログラムを使って、Pオプションを解除するには次のようにします。

① bload "pof"、&HF220, rで、pof プログラムを実行させて iset 命令を失かす

②Pオプションのかかっているプログラム(この場合は「ptest」)をロード ③ iset 命令を実行する

④ LISTをとることができる このように非常に簡単に済むので、他 のプログラムの解析に役立つでしょう。

●CRTの画面

とすると実行できます

```
bload "pof", 8Hf228, r -- 1
OK
load "ptest" -- 2
OK
list
Illey81 function call | UX + 0 E T US
SECTION |
18 FEW | 18 FEW | 18 FEW | 18 FEW |
18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW |
18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW |
18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW |
18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW | 18 FEW
```

Pオプションを解除せずに モニタに入る

このプログラムは先の ISET 命令のようにPオブションを解除しないで、モニタモードに入る方法です。モニタに入ったPオブションは解除されていませんから、BASIC モードに なればもちろん LIST はとることができません。

原理は簡単で、未使用命令の POLL 命令を再定義して F320H 番地に飛ばし、 そこでモニタモードに切り替えているだ

モニタモードにするためには、モニタ ROM が N-BASIC モード上にあるため に、I/O ポートの31H を切り替えます。

*モニタに入る方法

PUSH AF → A レジスタを保存する
DI ・和リ込みを禁止する

LD A. (DEGC2H) →前にI/Oボート3IHに出力したドータをもってくる

OR 04H → 2ピット目を1にする (2ピット目が1のときに N-BASICモードとなる)

OUT (31H) A +1/0 ボート31Hに出力する

LD (0E6C2H), A → I/O ボート3IH に出力したデータを次回聴ために保存する

EI →割り込みを終可する POP AF → A レジスタのデータを戻す

JP 6002H ーモニタ ROM のエントリポイントにジャンプする

プログラムとダンプリストは次のように なります。

■モニタに入るプログラム

: ***** P OPTION TEST ON MONITOR ******** : POLL COMMAND ON MONITOR : PROGRAM NAME ---> POM. e START: EQU @F328H E6C2 LOST: EQU 6E6C2H LOAD: EQU 6831 ERR2 MON: FOIL 68821 ORG START F328 3E28 LD A. 2BH (SEEASH) . A F322 32A8EE F325 3EF3 A. BESH F327 32A9EE (BEEASH), A F32B F5 PUSH AF F32D 3AC2E6 A. (IDST) F338 F684 OP RAH OUT (10AD), A

F339 C38268 JP MON

F33C END

■ダンプリスト

F334 32C2E6

F338 F1

F320 3E 2B 32 A0 EE 3E F3 32 A9 EE C9 F5 F3 3A C2 E6 F330 F6 04 III 31 32 C2 E6 FB F1 C3 02 60

PDP AF

(10ST).A

マシン語アラカルト

このプログラムも長くありませんから、 直接入力した方がよいでしょう。

プログラムの保存も先の「pof」と同じ ようにBASICモードにしてから BSAVE L # +.

BSAVE "pom", &HF320, &H2F1, 1

DUAD-88D条 使った場合

Foom | Foof | ≥ 6 € Nan-BASIC 07 ークエリアの領域を使用しているため に、DUADの Loader を使用した場合に は、他のディスケットに BSAVE できま せん、そこで、BSAVE する方法は、以下 の踊りになります。

① LOAD で [pom] または [pof] をロード ② Nan-BASIC のシステムディスケットを ドライブ1に入れて リセットボタンを 拥士

③モニタモードのDコマンドを使用して プログラムが壊れていないか確認 (4) BASIC モードに戻って BSAVE する

モニタに入って Pオプション解除

モニタモードに入ってしまえばPオブ ションを解除するのは楽たものです。 ① "pom" をロードする ②Pオプションのついたプログラムをロ ードする

●CRTの画面

bload "pof". &Hf228. ---

load "ptest"---2 Ok

PO11-(3)

hlsec28 -- 4 EC29 20-88-5 Ok

18 REM ****** P OPTION TEST ****** 28 PRINT " **** P OPTION TEST PROGRAM *** "

38 PRINT 48 PRINT " P x7" 582 7 NZ" Z *0*0 N 2090 7" Z "

SA PRINT PRINT " 2770 t popper 29"74."

③ POLL P を実行する

④モニタモードのプロンプトがでるの で、Pオプションフラグのある番地 (EC29H) を贈べる

⑤ EC29H 番地が20H となっているので、 00H を書き込む

⑥ BASIC モードに戻る、Pオブションが はずれているのでリストもとれる

いかがでしたか Pオプションの解除 のやり方はわかってしまうと非常に簡単 ですね、BASIC のコマンドだけでは絶対 にPオプションは解除できません こんなところにもマシン語の威力があ

るのです

à la carte 16

一気に移動させる ブロック転送命令

Z-80CPUには任意のバイト数のデータを別のアドレスへ移動させる命令がある

プロッタ転送というのは、目に、メモリトにあるデークを別の領域のメータ 別とないをとは側目とまっ、このプロック 転送命令は数ステップのプログラムを 自命をで実行し、スピードも強くよく利 用きれています。プロック転送命令は4 つ用意されており張力です。

ペアレジスタの用途は4つのブロック 転送命令に共通で、次のようになります。

■ブロック■送命令時の ペアレジスタの用途

HLペアレジスタ

――移動するメモリの番地

DE ペアレジスタ ---データの行き先の番地

BC ペアレジスタ

──バイト・カウンタ
■●■されるフラグ

P/V フラグ (Cy, S, Z フラグは変化しない)

1バイトごとのブロック転送 LDI命令

これは1バイトごとのブロッタ転送で す。[JP、JRJなどのシャンプ命令を用い てループを作ってブロック転送を行うも のです。自分でループを作ることができ ますので、ループの間に何かの仕事をさ せることができます。 ベアレジスタとフラグの動きは次のよ うになります

■ペアレジスタの動き

HL = HL + 1DE = DE + 1

BC = BC - 1

●フラグの動き

P/V フラグ

実行した結果、BC ベアレジスタが 0 の とき 0. 1以上のとき』となる

HL ペアレジスタが指すメモリのデータを DE ペアレジスタが指すメモリに転 道機 HL DE ペアレジスタは1を加えら れて、次の転送先の準備を行い、BC ペア レジスタから1を引きます。BC ペアレ ジスタに転送するバイト数を入れておけ ほ自動的にカウントしてくれます。

また、P/V フラグは BC ペアレジスタ がりになったときにりとなるので、この フラグを調べれば、BC ペアレジスタが りになったかわかります。それには [JP PO、nn]、[JP PE、nn]などの命令を使 州します。

VRAM の1行のデータを他の番地に 転送するプログラムを作りましょう。

VRAM —→ F3C8H

転送番地先 → D000H 転送バイト数→ 50H バイト

どうですか? LDI命令を使用すると

■ LDI を使用しなかったとき

LD HL, OF3CBH LD DE. GDOOGH

LD BC, 0050H

LOOP: LD A. (HL)

LD (DE), A

IND HI

IND DE DEC 8C

LD A. B OR C

JP NZ, LOOP

■ LDI を使用したとき

LD HL, DF3C8H LD DE. ODDOOH

LD BC, 0050H LOOP: LDI

JP PO. LOOP

非常に短くなることがわかると思いま

この LDI 命令は1バイトずつのブロ ック転送命令ですから、必ずジャンプ命 令と組み合せて使用する必要があります。

連続してデータ転送する LDIR命令

このLDIR命令は連続してデータ転

送をするものです。内部で自動的にルー ブをしてくれるので、ルーブに必要なシ ャンプ命令などは必要ありません。した がって、フラグを糊べる必要もありませ ん、使い方は LDI 命令と全く同じです が、ループが自動的なので、ループ間に せん

LDI 命令を使用したサンブルプログ

■ LDI 命令

LD HL, 0F3C8H LD DE. ODOODH

LD BC, 0050H LOOP: LDI

JP PO. LOOP

● LDIR 余全

LD HL. DE3C8H

LD DE, ODGOCH LD BC. 0050H

LDIR

大きい番地から小さい番地へ

ラムと LDIR とを比較してみましょう. このように、ジャンプ命令がない分だけ すっきりしたプログラムになります. LDI 命令よりもこの LDIR 命令を使用 することが多いようです。

データ転送~LDD命令 LDD 命令は LDI 命令と同じ1バイト

ことのデータ転送命令なのですが、大き

く違うことは HL, DE ペアレジスタとも に 1 を引かれることです。LDI 命令はア ドレスの小さい方からデータ転送するの に対し、LDD 命令はアドレスの大きい値 から小さい値へとデータ転送することに かります。

ベアレジスタとフラグの動きは次のよ うにかります

●ペアレジスタの動■

HL=HL-1 DE=DE-1 BC=BC-1

■他の命令では

LO HL, 0F417H LO DE, 0D04FH LD BC, 0050H

LOOP: LO A, (HL)

LD (DE), A
DEC HL
DEC DE
DEC BC

OR C

このようになります、LDI命令と違う ところは、HL、DEベアレジスタにセットされるアドレスが、大きい方のアドレ スで、小さい方のアドレスに向ってロードされていくことです。

BCレジスタが 0 になる まで転送~LDDR命令

LDDR 命令は連続してデータ転送を する命令です。内部で自動的にループを 作ってくれますから、BC ペアレジスタ

●フラグの動き

P/V フラグ 実行した結果、BCペアレジスタが 0 のとき 0.1以上のとき 1

これらの動作については、HL、DE ベ アレジスタが 1 になる他は同じです。 VRAM の1行のデータを他の番地に 転送するプログラムを作ってみましょ

VRAM -→ F3C8H~F417H 転送番地 -→ D000H~D04FH

転送番地 → D000H~D04FH 転送パイト数→→50H

■ LDD 命令を使用す III と

LD HL, 0F417H

LD BC. 0050H

LOOP: LDD

JP PO, LOOP

が 0 になるまで転送してくれます。 LDD 命令で使用したサンブルプログ ラムと比較してみます

この命令もジャンプ命令がない分だけ プログラムがすっきりしています。

● LDD 命令

LD HL, 0F417H LD DE, 0D04FH

LD BC, 0050H

LOOP: LDD

JP, LOOP

● LDDR 命令

LD HL, 0F417H

LD DE, 0004FH

LDDR

②転送元と転送先のメモリエリアが言なる (オーバラップする)

オーバラップする場合ははっきりと2 つのグループに違いがでてきます。つま り、どちらの方向に移動するかによって、 どちらか一方しか使用できなくなりま す。

この違いをはっきりと認識していない と、転送元のデータを壊すことになりま す

■ LDI, LDIR 命令が使用で⊪るとき。

図図のようにLDIと LDIR 命令が費用できるのは金字板送のアドレスが幅送で、 正なるのは全字板送のアドレスが幅送で、 らないことです。これは、板送先のアドレスが構造のアクタースリアのアドレス になっても、このときすでに、板送たの アータは新しいエリアに板送されている ので、壊されても良いからです。LDI、 LDIR 命令が使える条件は以下のとおり です

LDI、LDIR が使える条件

HLペアレジスタ、DEペアレジスタ のデータ

(転送元アドレス) (転送先アドレス) □ LDD、LDDR 命令が使用できるとき

図③のように、現在のデータエリアを アドレスの大きい方に転送するには、ア ドレスの大きい方から転送しないと転送 元のデータが壊れることが理解できると

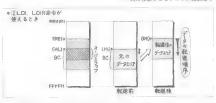
4つの命令を使い分ける

プロック転送命令には「LDI、LDIR、 LDD、LDDR」の4つの命令があります が、実際には2つのグループに分けるこ とができます、「LDI、LDIR」と「LDD、 LDDR」の2つです。

それぞれの使い方について、メモリエリアを例にとって説明しましょう。 ①転送元のメモリエリアと転送先のメモ



図①のように転送元のメモリエリアと 転送先のメモリエリアが重ならない (オーバラップしない) ときは4つの命令を 自由に使用できます。





思います。このように、アドレスの大き い方に転送できるようにしたのが LDD、LDDR命令です、LDD、LDDR命 令が使える条件は次のようになります。

LDD, LDDR 命令が使える条件 HL ベアレジスタ DE ペアレジスタの

のデータ データ (転送先アドレス)

プロック転送命令を使用するときには、どちらの方向に転送するかを調べて、それに合った命令を使用してください、特にこれらの命令を有効に利用できるのはCRTへの引き込みやクリアのときで、高速化が図れます。

à la carte 17

微妙な中間色も出せる アナログモード

SRの特長のひとつの512色出せるアナログ モードをマシン語で使うには

512色出すのに必■な ハードとソフト

SR の並人の特別の1 つにアナログモードがあり、512色中から、任意の色を 8 金選択できるすこい機能です。"せっかく マシン語で機能してみたいと思います。ここでは、アナログモードで512色を出してみましょう。ただし、512色出すためには、次 のハードウェアとモード設定が必要です。

●ハードウエア

アナログ RGB 用 CRT

●ソフトウエア

V2モードでのみ可能、ただし、グラフィックスモードがモノクロの場合は、 テキスト画面で使用できる

8レベルの明るさで 色512色

デジタルモードでのカラーは一般的に は8色までしか出せません。それは、 CRTがあ、高、線の色の組み行わせでで 8色色だけです。デジタル保砂は常に0 かち Vの 2 つに つという原則にした がっている関係に、2 V ラ 3 V という 間の電圧は存在することが許まれないか です。たなかって、カラーCRTでも常 に基色が光っているか、光っていないか のどちらかしかないわけです。ですから 3色の組み合わせでできる色は8億まで ということになります。RGB専用CRT ほそれぞれの3色がよく出るように設計 されていてるために、アナログモードの ように健用したいと思っても無限です。

それに対してアナロタモードでは発光 総がデジタルモードと同じる原色しかないのですが、それぞれの色の明るさを電 IFの人きさで、コントロールすることによって、多くの色を刊せます、SRのアナ リケモードでは各色の電性をまり、CRのアナ ときまでコントロールできるので、各 色はそれに応じて8人へルの明るさきも つことになります。したかって、

赤緑屬

8レベル×8レベル×8レベル=512色 が出せるわけです。

それぞれの色のレベルはミレベルです から、2進数で表すと次のようになりま す。

8レベル=23ビット

8レベルの明るさ (難度という) をコ ントロールするためには、それぞれの色 に3ビットですから、全部で9ビット必 要になるわけです。

しかし、この9ビットは少しやっかい なビット数です。8ビットより1ビット 多いために2パイト必要となります。そ こで、SRでは、mkII などとの互換性を 考え、パレット I/O ポートに II 回声き込 むという方法をとっています。

少しやっかいなのですが、実に便利な 使い方もあります。1 つのパレット1/O で512色出すことができるのです。このパ レット1/O はデジタルモードと同じ1/O アドレスを使用していますので、同時に 出せるのは512色中8 色までです。デジタ ルモードのカラーパレットは、3 ピッ ト・データのみ書き込んでいましたが、 アナログモードのカラーパレットは、 ビットにデータを書き込めるので512色 より海線が可能になったのです。

カラーパレットの 設定のし方

バレット番号と1/0 アドレスのデー タは同じです。しかし、デジタルモード とアナログモードではカラーモードの設 定が違うので注意が必要です。

●デジタルモード時のカラーパレットの設定

1/0	パレット番号	カラーデータ(初期設定時)	Ē
54H	0	-000	黒
55H	1	0 0 1	zale: PS
56H	2	- 0 1 0	赤
57H	3	0 1 1	紫
58H	4	- 1 0 0	緑
59H	5	1 0 1	水色
5AH	6	-110	黄色
5BH	. 7	111	白

このビット 一印は使用しない で決まる

●アナログモード時のカラーバレットの設定

	パレット	カラーデータ 1			カラーデータ 2							ê			
ポート 番 号	7 6	5 4 3	2	1	DE)	7	6	S	4	3	2	;	Brisk		
54H	0	0 1		0	0	0	0	0	0	0	D	0	0	0	50
55H	1	0.1		0	0	0	0	0	0	0	0	1	1	1	青
56H	2	0.1		0	0	0	0	0	1	1	1	0	0	0	赤
57H	3	0 1		0	0	0	0	0	1	1	1	1	1	1	紫
58H	4	0.1		1	1	1	0	0	0	0	0	0	0	0	綠
59 H	5	0.1		1	1	1	0	0	0	0	0	1	1	1	水色
5AH	6	0.1		1	1	1	0	0	1	1	1	0	0	0	黄色
5BH	7	0 1		1	1	1	0	0	1	1	1	1	1	1	白



例 2 01000100B…中間の明るさの緑

てしまいます.

もし、パレット番号1に赤っぽっい黄 色を出すのであれば、

LD A, 38H (001110008) ← OUT(55H),A 赤を7レベルの間度に

A, 44H (01000100B) -

OUT(55H)、A 最をもレベルの原理に というようになります。自分で好きな色 が出せますが、CRT上の色と基準色とは ずい分と違いがありますので、一度全部 CRT上に表示してから決めるといいで

テキストモードで512色出す プログラムを作る

実際にどのような色が出るのか、プログラムを作ってみてみましょう。このデモプログラムは色を見るためのものですからカラーに関係ない部分は簡単にします。このプログラムのフローチャートはP.287のようになっています。

色を変えていく方法はカラーデータの 1とカラーデータの2から順番に1を引 いています

アナログモードにするには I/O ポートの32H にデータを参き込みます



m フローチャート



1/0 ボート32H の5 ピット目と6 ピット目はアナログモードが使用できるか、 できないかを決めているスイットです。 他のピットは ROM のペンク切り形える どで使用していますので、勝手に変化させることはできません。このボート32H は IN 命合が実行できますので、現在の モードを知ることができます。 マシン語で5 ピット目と6 ピット目だけ1 にするには次のようにします。
IN A. (32H) - 現在のデータを読み出す

OR 60H ・・60H を加える。他のビットは変化しない

OUT(32H), A・ ちじっト目とまピット目

もし、送ったデータが必要であれば、 PUSH AF, POP AF で保存しておいて もよいでしょう

なお、このテキスト画面のアナログカ ラーは、グラフィックス画面がモノクロ時 のみ有効ですから、プログラムを実行す る側に必要状の命令を実行してください。 SCREEN I, 0, 0, 7○○

OK

なお、このプログラムは実行し終わる までに約4分10秒かかります。 プログラ ムは P.228にあります。

もう少し色を ゆっくり見たいとき

もう少し色をゆっくりと見たい場合 は、WAIT ルーチンのデータを大きくし てください、B980H の20H を大きくすれ ばよいのです。

また、このプログラムに多少手を加え て、カラーパレットに送っているデータ を表示させると便利になります。 D. E レジスタがそれぞれの色のデータをもっ ているので、DISPのサブルーチンの中 で DE レジスタのデータを表示させるよ うにします。数パイトで済みます。

バックグランドも 512色出せ

アナログモードでは、バックグランド の色も512色使用できます。色の出し方

●アナログモード時のバッ クグランドカラーの設定

1/0ポート	設定する色のレベル	,	カラーテ 5 5 4 3		OE A
54H	BLUEのレベル	1	0	××	×
54H	REDのレベル	1	0 ,× × ×		
54H	GREENのレベル	1	1	××	×

ここは必ず1でなければならない

は P.285のテキストのアナログカラーと 会 (同じですが、カラーデータを設定す も I/O ボートがデジタルモードとアナ ログモードとでは違います。チジタルモ トでは52H でしたが、アナログモード は54H になっています。この54H はバ レット機能で使用されています。アピット HI を1にすることによって、バックグ ランドのカラーデータかパレットのカラ データかを代拠しています。 ボートはパレット機能のみですから、絶対に7ビット目を1にしないでください。もし、出方した場合には動作が保証されません

デモプログラムは P.287と同じですか ら理解できると思います。

アナログモードの512色は比較的に美 しいので、いろいろな、コンピュータグ ラフィックスなどにおおいに使用できま す。グラフィックス時のバレット のデールをといった。

1760	7777	7-900	200	100777	フフィックスなどにおおいに使用でき.
テー	タかを XS	祖してい	主す		す。グラフィックス時のパレットのデ
ただし	.,他の55	H から5	BH #	での1/0	夕設定のやりかたは前と同じです。
512色	を順次表	示するフ	ログ	ラム	
			***	*****	************************
		PROGR	AM MA	NE> ANLG	PC6881 MK2 SR
		TEXT	/ 119-	J 5" E7" D2" 98	(7702° E-F°)
			****	*********	***********************************
B989 E626 F3C8 6F6B 885B 3E9B		START: MON: ABCRT: COLOR: PALT: BELL:	EQU EQU EQU	88988H 8E626H 8F3C8H 6F6BH 5BH 389BH	:CRT COLOR SUBROUTINE :ANLG PALET 7 :BEEP SOUND
		1	ORG	START	
		CRT M	ODE P	ROGRAM	
	8E19 3E81 32B2E6		LD LD	B.50H C.19H A.81H (@E6B2H),A	:3コ / ケタスウ サ セット トル 代くBレジ"スタン8名 :タテ / キ"ヨウ スウ ヲ セット スル 8くCレジ"スタン2 :スクロー目 カイシ キ"ヨウ
BBBE	32B3E6		LD LD	A.19H (0E683H),A A.8E8H (0E684H),A	; スクロール シェウリョウ キトョウ : カラーコート
B913			1.D	A, BBH	1772/93 3-1" ON.01H OFF.80H

微妙な中間色も出せるアナログモード

			微妙な中間色も出せるアナログモー
B915 32B8E6 B918 3EFF B91A 32B9E6 B91D CD6B6F	LD LD LD CALL	(@EGB8H).A A.@FFH (@EGB9H).A COLOR	:#9- マ-ク #9FFH モノクロ,88H
	: MAIN PROG	RAM	
B928 CD8B42	MAIN: CALL	428BH	: CURSOR OFF
B923 DB32 B925 F660 B927 D332	OR OUT	A.(32H) 6#H (32H).A	; アナログ キード ニスル
B929 1647 B92B 7A	LOOP2: LD	D. 47H A.D	:GREEN LEVL
B92E 1E3F B930 7B B931 D35B B933 CD5DB9 B936 CD7DB9	LOOP1: LO DUT CALL CALL	E.SFH A.E (PALT).A DISP	: RED. BLUE LEVL
B939 1D B93A C238B8 B93D 7B B93E D35B B948 15	LOOP1: LD LOOP2: LD LOOP1: LD LOOP1: LD LOOP1: LD LOOP1: LD LOOP1: LD LOOP1: LD CALL DEC JP LD OUT DEC LD CCP JP :	E NZ.LOOPI A.E (PALT).A D	
8941 7A 8942 FE48 8944 D22889	CP JP	48H NC,L00P2	
B947 CD9B3E	CALL	BELL	
894A 3E3F 894C D35B 894E 3E47 8950 D35B	CALL. LD OUT	A.3FH (PALT).A A.47H (PALT).A	: モニター ニ モトトル トキ ノ カラー シロ
B952 CD9842	CALL	4298H	: CURSOL ON
B955 3E8C B957 CD1608	LD CALL	A.8CH 8818H	:CRT CLEAR DATA 8CH :CRT CLEAR SUB
B95A C326E6	JP	MON	
	DISP SUB		
B95D E5 B95E D5 B95F C5 B968 F5 B961 21C8F3 B964 112888 B967 BF19	DISP: PUSH PUSH PUSH PUSH LD LD LD LD DLOP2: LD INC DEC JP	HL DE BC AF HL.ADCRT DE.8828H	
8969 8658 8968 3EE9 896D 77 896E 23 896F 85 8978 C26DB9	DLOP2: LD LD DLOP1: LD INC DEC JP	B.58H A.#E9H (HL1.A HL B NZ.DLOP1	:DISPLAY CHR *

マシン語アラカルト

```
B973 19
                          ADD HL.DE
B974 €D
                          DEC C
 8975 C269B9
                               NZ.DLOP2
                          POP
                               AF
 B978 F1
                          POP BC
POP DE
 B979 CI
 B97A D1
                          POP
                               HL.
                          RET
                  :WAIT SUB ......
 B97D C5
                          PUSH BC
                  WAIT:
B97E F5
B97F #E#2
                          PUSH AF
LD C,82H
                         LD
LD
B981 86FF
B983 3EFF
B985 3D
                  WL3:
                  WL1:
                          DEC
                          JP
                                NZ.WL1
                          DEC
 898A C283B9
                          JP
                               NZ, WL2
898D 8D
898E C28189
8991 F1
                          JP
                                NZ.WL3
                          POP
                               AF
                          POP
                               BC
 B993 C9
                          RET
B994
                          END
```

		*******	在食品 化氯化甲基 化异子基甲基 医克克斯氏 医克克斯氏 医克克斯氏 医克克斯氏 医克克斯氏
	PROGRAM	MANE> ANLG	B PC8801 MK2 SR
	TEXT / 1	カーノ デ [*] モフ*ログ [*] ラ	ム (パンッククンラントンノカラー)(アナログ、モート))
		**********	法事本的政治政治政治政治政治政治政治政治政治政治政治政治政治政治政治政治政治政治政治
B9## E626 F3C8 6F6B ##54 3E9B	START: EG MON: EG ADCRT: EG COLOR: EG BAK: EG BELL: EG	0U 8E626H 0U 8F3C8H 0U 6F6BH	:CRT COLOR SUBROUTINE :ANLG BACK GRAND COLOR :BEEP SOUND
	OF	G START	
	CRT MODE	PROGRAM	
8988 6658 8982 8E19 8984 3E01 8986 3282E6 8989 3E19 8988 3283E6		C.19H A.81H (8E6B2H).A A.19H (8E6B3H).A	1200-8 302 4° a0 1200-8 9200a0 4° a0
898E 3EE8 8918 3284E6	Lt	(8E6B4H).A	(カラーコート"
B913 3E00 B915 32B8E6	LD	(BE6B8H), A	:7>>29> =- 1 ON.81H OFF.88H
B918 3EFF B91A 32B9E6	LD		: カラー マーク カラー、FFH モノクロ、80H

微妙な中間色も出せるアナログモード

B91D	CD6B6F		CALL	COLOR		
		MAIN	PROGI	RAM		
B928	CD8B42				; CUR	SOR OFF
B925 B927	DB32 F66B D332		TUG	68H (32H).A		ログ゛モート゜ニスル
892B 892C 892E 893B	16C7 7A D354 1EBF 7B	LOOP2:	LD LD OUT LD	D.#C7H A.D (BAK).A E.#BFH A.F	: GRE	EN LEVL
B931 B933 B936 B937 B938 B93A B93D B93E	D354 CD5589 1D 7B FE86 D23889 78 O354	00011	OUT CALL DEC LD CP JP LD OUT	D.8C7H A.D. (BAK), A E.8BBH A.E. (BAK), A MAIT E.E. (BAK), A MAIT E.E. (BAK), A MAIT E.E. A.E. (BAK), A MAIT E.E. B.E. (BAK), A MC.LOOP1 A.D. A.D. A.D. B.C.B.MC.LOOP2		
8948 B941 B942 B944	7A FEC0 D22BB9		LD CP JP	A.D BCBH NC.LOOP2		
B947	CD9B3E		CALL	BELL		
	CD9842	:	CALL	4290H	; CUR:	SOL ON
B94D B94F	3E8C CD1888	;	LD CALL	A. ECH 8818H	CRT	CLEAR DATA 8CH CLEAR SUB
B952	C326E6		JP	MON		
		WAIT :	5UB			
8955 8956	CS F5	WAIT:	PUSH	BC AF		
8959 8958 8950 895E 8961 8962 8965 8965	#282 #6FF 3EFF 3D C25DB9 #5 C25BB9	WL3: WL2: WL1:	LD LD DEC JP DEC JP DEC JP	BC AF C.02H B.8FFH A.8FFH A.7FFH A.7FH B. WL1 B. WL2 C. WL2 C. WL3 AF BC		
B968	C9		RET	BC		
B96C			END			

à la carte 18

キーワードと 中間言語を見る

Noo-BASICのキーワードと中間言語を見る にはどうしたらよいか.

BASICコマンドは 中間言語に変換される

BASIC インタブリクは、BASIC のコ マンドを必ず1バイトから2バイトの BASICインタブリタ専用の言語に変換し てから実行しています。インタブリタ専 田の主頭を一般的に中間言語と呼んでい ます。同じCPUであってもBASICインタ プリタが違えば中間、(語も違ってきます。

この中間言語は表には出てきませんが、 BASIC の構造を測べる上では役に立つご 語です

また、この中間言語をうまく使用した コンパイラも発売されています。一般的 に、Pコードコンパイラなどといいます。 有名なのは、「UCSD PASCAL」などの P.SVSTEM CT PI-KIDNKIF は完全なコンバイラではありませんので、 インタブリタ上り渡いという程度です キーワードと中間言語のコードをプリ

ンクに印字する方法を考えましょう。 中間言語とキーワードの 格納番地

中間言語とキーワードを調べるには、 ROM Lのどこに、どのように格納され ているかを知る必要があります。 中間言語とキーワードの分類は、キー

ワードの1文字をアルファベット順にし ています

この分類のやり方はまつの方法によっ て行われています。まず。キーワードの 1文字目を調べて1文字目と同じアルフ ァベットを使用しているグループに分け ます。今度は同じグループ内で2文字目 口降の文字を測べてキーワードを区別! ています (P.293の図を参照のこと).

A~7.グループのアドレスは次のよう になっています。

A-688AH 0-604FH B - 6BA0H P - 6D68H C -- 6BAEH 0 - 6D97H

D - 6COEH M - 6D98H E - SCAAH II - SDDAH F -6C6FH T-6E21H

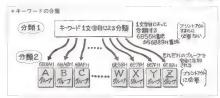
G 6C89H U - 6F41H H = 6C9BHV -- 6E4AH

I - SCA4H W -- SES8H J SCCEH X - SE7BH K -6CCFH Y -- 6F7FH

1 -6CDCH Z -6E80H

M-6D1CH N -- 6040H

たとえばCを頭文字とするグループは 6BAFH 番地からあります。もし、"COPY" というキーワードの中間は語を知るうと 思ったたらば GRAFH~6CODH の中にあ



るということです

はよくセパレータとして使用されていま 4

どのように 格納されているか

キーワードと中間言語が格納されてい る番地はわかりましたが、これだけでは 中間言語表をプリントアウトすることは できません。キーワードと中間言語がど のように格納されているかを知る必要が あります では、どのように格納されて いるかを調べてみましょう。

下のダンプリストは、6B8AH番地から をプリントアウトしたものの一部です。 6B8AH 番地は、A を頭文字とするグル

一プが入っている番地です。一番初めの 多グループと各グループの間には、00 キーワードは"AUTO"なのですが、これ Hというデータが書かれています。00H ではよくわかりません。しかし、データ 構造がわかると、すぐ区別がつきます。 キーワードと中間言語の順番は、次のよ うになっています

これを見るとすぐわかるように、キー ワードの後にすぐ中間言語がきています。 次のキーワードとの間にも区切りを示す 記号は何もありません。 中間言語コード は1パイト構成と2パイト構成がありま す。しかし、この ROM の中では2パイ トのコードを強制的に1バイトで表して います

では、1パイトコードと2パイトコー ドの格納のされ方を見てみましょう。

●6B8AH雲地からのダンプリスト

688A 55 54 CF A8 4E C4 F8 42 D3 86 54 CE 8E 53 C3 15 UT74NF BE T# S7 689A 54 54 52 A4 EB 68 53 41 56 C5 D5 4C 4F 41 C4 D4 TTR. 4 SAVIZLOAFT 68AA 45 45 D8 D7 88 4F 4E 53 4F 4C C5 9D 4F 58 D9 CD EEE5 ONSOLT OPAN 68BA 4C 4F 53 CS CB 4F 4E D4 99 4C 45 41 D2 92 53 52 LOSTPONT-LEAM SR 49 CE 54 49 4F D4 IC 53 4E C7 LISTING SNY DRO ID 44 42 CC 1E 6BDA 56 C9 28 56 D3 21 56 C4 22 4F D3 8C 48 52 A4 16 V/ VEIVE OF HR. 6BEA 41 4C CC BI 4F 4D 4D 4F CE ALTYOMMOSTHALS=0 BS 48 41 49 CE B7 4F 88FA CD 5A 49 52 43 4C C5 CC 4F 4C 4F D2 CB 4C D3 CF 5Z1RCL#70L0xbl#k

■中間胃節コード 1パイトの場合 キーワード A U T O A8←中間言語コード 1 1 1 1 理想的なデータ 41 55 54 4F A8 1 1 1 1 55 54 CF A8 実際のデータ なし

"A"というコードが入っていません。 これは分類の1でキーワードの顔文字で 分類1. ているので特に必要ないからです。 は 次のデータが中国 注語コードである また、"O" のアスキーコードが、4FH→ CFH になっています。キーワードはすべ てアルファベットからできているので. アスキーコードはすべて、7FH以下のデ ータになります.

そこでアルファベットのアスキーコー ります ドでは使用していない7ビット目(1× ××××××B) を無理に使用している のです。つまり7ビット目が1であるか ないかによって、キーワードの終りの目 のデータが必ずFFHとなっており、意味 印に使用しているのです。キーワードの 長さはさまざまでどこでキーワードが終 っているか区別がつかないので、キーワ ードの終りをこのビットが1になってい るかどうかだけで知ることができるので す。本当ならば00Hなどのセパレータを 1 バイト入れておけばいいのですが、メ

モリを節約するために工夫しているので す したがって、7ピット目が1のとき ことを表していることになります

また 中間 3 語コードが1 バイトなの で、7ビット目が1になっているデータ からりつ生のデータが 次のキーワード の2文字目から始まっていることがわか

中間言語コードは2パイトで表されて いるものもあります。中間言語コードが 2パイト構成になっているものは、初め があるのは2番目のデータだけです で は「ABS」命令を例にとってみましょう。 キーワードの表し方は、「AHTO」と全 く同じです。しかし、中間言語コードは 1バイトで表されています よくみると このバイトで表しているデータも、加工 されていることがわかります 2パイト

*中間常開コード 2バイトの場合 *-7-F A B S FF 86←中間言語コード 1 1 1 理想的なデータ 41 42 53 FF 86 実際のデータ 42 D3 06 なし (86H - 80 = 06H)

の中間言語コードを11イトで変すため に、アピット目を強制的にのにしている のです。それによって、表される11イイ トの中間言語は常に80日よりも小さいと いうことになります。したがって、80日 よりも小さい場合は2イイトの中間言語 ということになります。このようにして、 ここでもメモリの節刻をしています。

プリントアウト するために

キーワードとその中間言語をブリンタ にブリントアウトするためには、次のよ っかことを知る必要があります

- a. プリンタにデータを出力する方法 b. 16進数を 2 桁の ASCII コードに直す 方法
- まず、この2つの方法について説明しましょう。

握手によく似た ハンドシェーク法

CRT 上にTEXT 文字を表示させるには、VRAMにアスキーコードを書き込めはなかったのでが、プリッタルジプリントアウトするには、プリッタのイシターフェースにデータを送らなければなりません、送るタイミングをしっかりわせてあげないと、データを送ってもプリッタに顕彰いてくれません、データをプリンタに確定に検送する方法をハンドシェーク

法と言います。ハンドシェーク法はデー タ8ビット以外に2本の信号線をもって おり、この2本の信号線でデータを確実 に送っているのです。ハンドシェーク法 の動きを簡単に説明しましょう。

まず、本体がプリンタにデータを送り たいと思ったら、データボート(I/O 10H) にデータをセットします。次に「今セッ プリンタに信号を送ります(I/O OUT 40 H 0 ピット ストローブ信号) この信号(ス トローブ信号) をプリンタが受けとり、 本体からのデータを読みとります。プリ ンタは既に読みとっていたり、今読みと ることができない場合は、本体に「今テ ータを送られてもデータを読みとれない よ」という信号 (I/O IN 40H 0ビット ビジー信号)を送り返します。本体側 は次のデータを送るときはビジー信号が 出ていないことを確めてから、またデ 一夕を送っていきます。このように 2本 の信号をお互いに送り合って正しいデー タの転送をしているのです。この2本の 信号線は人間の提手によく似ていますの で、このような名前がついています。

しかし、このような動きをするプログ ラム (ハンドラという) は自分で作るよ りは、ROMルーチン内にありますので、 そのルーチンを利用した方が便利です。 ブリンタに1 文字を送る ROM ルーチ ンは次のようになります。

*プリンタに1文字送るROMルーチン

- LD A, 41H ←プリンタに送るデータを A レジスタにセットする CALL DOT8H ←プリンタに 1 文字を送るルーチンを呼ぶ (RST 18H)
 - 注 プリンタスイッチがE64CHにあります。ここにODH以外のデータを書き込みます。もし、ODHならばCRTに出力します。

プリンタに1文字を送って印字するサ ンプルプログラムは次のようになります。

ID A 20H ←プリンタ出力モードにする LD (0E64CH), A ID A 41H ← A コードをセットする CALL 0018H LD A, OAH

←ブリンタのヘッドをその行の初めに戻すコードを

プリンタに送る CALL 0018H

LD A. ODH ←改行させるコードをプリンタに送る CALL 0018H

RST 38H ←モニタに戻る

このプログラムを実行すると、"A"と いう文字を印字し改行します。OA やOD (はコントロール・コードですが、このコ - ドをプリンタに送らないと、プリンタ は受けとったデータを印字しません。

16道 敷を2桁の アスキーコードに

キーワードはアスキーコードで格納さ れているので問題はありませんが、中間 言語は16准数なのでそのままプリントア

ウトすることはできません。そこで、ブ ログラムで16進数を2桁のアスキーコー ドにしなければなりません

プログラムは下のようになります。こ のプログラムではBレジスタに上位の桁 をセットしCレジスタに下位の桁をセッ トして厚ります

このように、上位4ドットと下位4ド ットを分けてアスキーコードを作ってい きます、A8Hを例にとって説明するとP. 297のようになります。

●16進数を2桁のアスキーコードにするプログラム BA11 F5 ASCII: PUSH AF - 下位4 ビット明ために保有 RLCA BA12 07 BA13 87 RL.CA 右シフト命令、上位すどョトを 1ビット目・11ビット目にシフトする BA14 87 RICA BA15 07 RLCA. BFH 一現在の上位4ビットを消す BALS EGGE AND BA18 CD24BA CALL AS B.A ←Bレジスタにコピー BAIR 47 L.D POP AF 一保存していたデータをとり出す BAIC FI BFH · トウ 4 ビットを選す BAID FASE BAIF CD24BA CALL AS BA22 4F LD C.A ←Cレジスタにコピー BA23 C9 RET BA24 FERA AS: CP でATI C、AD 4ビットが9より大きいか調べる BA26 DA2BBA Jp ADS BA29 C687 A、B7H、 9より大きければ07Hを加えて A、30H A-Fのデータにする BA2B C630 AD: RET 下位 4 ビットにO3H を加えてアスキーコードにする

MA8H			
①右へ4ビットシフトし、上位		2 進数で表すと	
4 ピットを消す	OAH	00001010B	
②9より大きいので07を加える	→OAH	00001010B	
	+)07H	+)00000111B	
	11H	00010001B	
③11日に30日を加えると、41日	11H	00010001B	
というAのアスキーコードに	+)30H	+)001100008	
なった	41H	01000001B	
④もとのデータを読み出して,			
上位 4 ビットを消す	→08H	00001000B	
⑤9より小さいのでそのまま30		00001000B	
H を加えると、38H というⅡ	+)30H	+)00110000B	
のアスキーコードになった	38H	001110008	

2つの 注意事項

このプログラムを作るときに注意しな ければならないのは、キーワードの頻文 字がすべてないことです。したがって、 それぞれのグループに頻文字をつけてや ム必要があります

また印字するときに、印字用バッファ を1キーワード分作っておき、次のキー ワードに入る前に印字をします。中間言 語のコードの印字位置を合わせるために キーワードの次に00日を入れておき、ブ リンドアウトするルーチンがパッファ内 900日を見つけると、ブリンタのヘッド を10桁目へもってくるようにしています。 パッファ内の終りの記号として00日を使 っており、00日を見つけるともとのプロ グラムに戻ります。

なおプログラムについては、リストに 説明してありますのでそちらをみてくだ さい、リストと実行結果を示します。

●キーワードと中間質額を見るプログラム

	KEYWORD LL	ST
	PROGRAM NA	ME> KW.e
	*******	***************************************
B988 8816 6B8A	PNT: EQU	8818H 8818H 582AH
	ORG	START

```
B988 ED7383BA
                    LD (SPSAVE), SP
8964 31A3BA
                    LD SP.STACK
                     LD 4.28H
B987 3E28
                                            PRINTER ON
8989 324CES
                          (8E64CH).A
B96C CD58BA
                     CALL CR
B98F DD2163BA
                         IX.PSC
B913 218A6B
B916 1641
                         HL. KWSAD
                          D. 'A' . 初めのキャラクタコードをセットする
B918 DD7200
             LOOP:
891C FE59
                          59H
                                  Yボキャラクタになったら、酒算子を
                     JP
                          Z.OJP3 ブリントアウトす用プログラムへとばす
B91E CAB6B9
8921 7E
                          A. CHL1
B922 FE00
                          88日 - 次のクルーフか馴べる
                    JP
8924 CAECB9
                          Z、ZERO · · · · · 次のグループ妨押プログラムへ
                          BBH 8DHより大きいときは、中間質器コードのセパレータ
NC、DJP 1 も含んでいるのでそのプログラムへ行く
8929 D235B9
                     JP
B92C DD23
                     INC
                         1X
                          ([X).A プリンクバッファに答える
892E DD7788
B931 23
                     INC
                         HL - 次のアドレスにする
B932 C31BB9
                          LOOP
B935 E67E
              0.1P:
                     AND
                          7FH・ーセバレータを切り離す
B937 DD23
                    INC
                         TX
8939 DD7788
                    LD
                          (IX).A
                                            SEPELETER STY & C. D.C.
893C 3E08
893E DD23
                    LD
                          A. BBH
                     INC
B948 DD7788
                     LD
                          (IX).A
B943 23
                     INC
                         HL
B944 7E
                          A.(HL) 中間含結が80Hより大きいか調べる。小さい
                          88日

GroJP2

場合は8パイトコードの処理するプログラムへ
8945 FE88
8947 DA7589
                     JP
B94A CDFFB9
                    CALL ASCII--中間言語をアスキーコードにする
B94D 75
B94E DD23
                     INC
B950 DD7700
                    LD
                          (1X).A
B954 DD23
                    INC
                    LD
8959 3E0D
                          A. ADH · 一印宝するとAに終りを示すセバレータ
895B DD23
                    INC
B95D DD7700
                    LD
                          (IX).A
B968 CD1CBA
                     CALL
                         PRINT
8963 23
                         HL
B964 7F
                          A, (HL)
B967 CA1BB9
                          Z.LOOP
B96A DD2163BA
                          IX.PSC· -バッファをクリアする
896E 7A
896F DD7788
                         (!X).A | 次の文字をバッファに送る
8972 C31889
                    LD
B975 3F46
              OJP2:
                          A.46H
B977 DD23
                     INC
                         TX
8979 DD7788
                          (1X).A
B97C DD23
                     INC
897E DD7786
8981 3E28
                          A,20H · - スペースを空ける
B983 DD23
                     INC
8985 DD7788
                          (IX).A
8989 F688
                          89H ----80H を加えてもとの中間言語コードに戻す
```

```
B988 CDFFB9
                  CALL ASCII
                   LD A.B
INC IX
LD (IX),A
LD A.C
B98E 78
B98F DD23
B991 DD7700
B994 79
                    INC IX
B995 DD23
                    LD (IX).A
B997 DD7788
B99A 3ERD
                    LD A. BDH
                   INC IX
LD (IX).
CALL PRINT
B99C DD23
B99E DD7788
B9A1 CD1CBA
B9A4 23
B9A5 7E
B9A6 FEB8
                    INC HL
                         A. (HL)
B9A5 FEB8
B9A6 FEB8
B9A8 CA1BB9
                    CP
                        BBH
                    JP Z.LOOP
LD IX.PSC
B9AF 7A
B9B8 007788
                    LD A.D
LD (IX).A
JP LOOP
B9B3 C31BB9
B9B6 DD2163BA OJP3: LD IX.PSC - - 演算子をプリントアウトする
             LD A, (HL)
7E
8988 Etr.
8980 DD7700
                    AND 7FH -セパレータを切り離す
                     LD (18).A
                                            SEPELETER
                     LD
                          A.BBH
                     INC
                         IX
B9C2 DD23
B9C4 DD7788
                     LD
                         IIXI.A
B9C7 23
                     INC HL
BACK 7F
                    L.D
                         A. (HL)
B9C9 CDFFB9
                 CALL ASCII
89CC 78
89CD DD23
89CF DD7700
                    LD A.B
INC IX
Z.STOP ← 全部印字したらストップ
LOOP
                    JP
 B9E9 C31BB9
          ZERO: INC
                         ■ ・ 次の頭文字にする
 BSEC 14
                         IX.PSC
 B9ED DD2163BA
                     LD
 B9F1 DD7286
                     LD
                     INC
                         HI.
                     JP
 89F5 C31BB9
                         LOOP
 B9F8 ED7B83BA STOP: LD 5P.(5PSAVE)
B9FC C33888
                     JP 8838H
                                           : GOTO MONITOR
 B9FF F5
             ASCII: PUSH AF
                                           :ASCII CODE
 BA00 07
                     RLCA
 BA81 87
BA82 87
                     RLCA
                     RLCA
 BA03 07
                     RLCA
 BAR4 ESSF
                    AND REH
 BABG CDI2BA
                   CALL AS
                     LD B.A
 BA69 47
 BARA F1
```

マシン語アラカルト

```
BARR ERRE
                     AND ØFH
BARD CD128A
                     CALL AS
                    LD
BAIR 4F
BALL CS
                     RET
BAL2 FE0A
              AS:
                          BAH
BA14 DA19BA
                     JP
BA17 C607
                     ADD
BA19 C638
              AD:
                         A.38H
BAIC DD2163BA PRINT:
                         IX.PSC
                                      : PRINTER OUT
BA28 LESS
                          E.88H · Fレジスタでブリンタヘッドの分階をカウント
BA22 DD7E00 PL2:
BA25 FE00
                          BEST
                     JP.
BA2A CD1888
                     CALL PNT
                     INC
BA38 C322BA
                         PI.2
BA33 SEBA
BA35 93
                         B.A
                                  中間登録コードが10桁目になると
BA37 3E2B
                          A.28H
                                  計畫1.至20日を出力する
BA39 CD1800
                     CALL
BAGC 85
BASD C239BA
                     JP
                         NZ.PL4
BA48 DD23
                     INC
                                  ODH (終りの印) がく当までバッファ
BA45 FEED
                         BDH
BA4A CDI888
                     CALL PNT
BAAD C348BA
BASE CDS8BA
                     CALL CR
                     LD
                         IX.PSC
                    RET
BAS8 3E0D
                    LD A. BDH
BA5A CD1868
BA5D 3E8A
BA5F CD1868
                    CALL PNT
                    LD
                    CALL PNT
BA62 C9
                    RET
BA63
              PSC:
                   DS
                         28日 --- 1 行信文字を書えるバッファ
BASS
              SPSAVE: DS
                         82H
                         38
BAAS
              STACK:
BAA3
                    END
```

■風行機 ■(中国第四コード)

AUTO	A8	CONSOLE	9D	CV1	FF A8
AND	F8	COPY	CD	CVS	FF A1
ABS	FF 86	CLOSE	CB	CVD	FF A2
ATN	FF 8E	CONT	99	COS	FF 8C
ASC	FF 95	CLEAR	92	CHR5	FF 96
ATTRS	EB	CSRLIN	FF D4	CALL	B1
BSAVE	DS	CINT	FF 9C	COMMON	B6
BLCAD	D4	CSNG	FF 90	CHA1N	B7
BEEP	D7	CDBL	FF 9E	COM	FF DA

キーワードと中間言語を見る

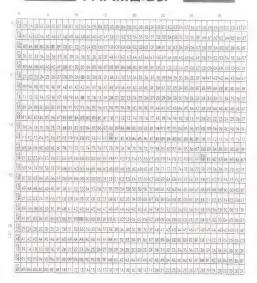
CIRCLE COLOR CLS	CC CB CE		LLIST LPOS LET LINE	9C FF 88 AE	98	RENUM RANDOMIZE ROLL SCREEN	A9 B9 D8	
CMD DELETE DATA DIM DEFSTR	FF A7 84 86 AA	E4	LOAD LSET LIST LFILES	C1 C6 93 C9		SEARCH STOP SWAP SET	FF 98 A2 BF	DG
DEFINT	AB		LOG		8A	SRQ	800	-
DEFSNG	AC		LOC LEN	FF	A4 92	SAVE	FF C8	E3
DEFDBL	AD BA		LEFTS	FF	81	SPC1	E2	
DEF	97		LOF		A5	STEP	DF	
DSKIS	EC		MOTOR	FF		SGN	FF	84
DSKF	FF	DB	MERGE	C2		SQR	FF	87
DATES	FF	D9	MOD	FD		SIN STR\$	FF	93
ELSE	9F		MKI\$ MKS\$	FF	A7 A8	STRING\$	ES	93
END ERASE	A3		MKDS	FF	A9 :	SPACE\$		98
EDIT	A4		MIDS	FF	83	THEN	DD	
ERROR	A5		MON	CA		TRON	AB	
ERL	E4		MAP	FF 83	DS	TROFF	A1 DE	
ERR	E5	0.0	NEXT	C4		TO	DC	
EOF	FF	AS AS	NEW	94		TAN	FF	8D
EQV	FB	No.	NOT	E3		TERM	02	
FOR	82		OPEN	BB		TIMES	FF	DC
FIELD	BC		OUT	9A		USING	E7 EØ	
FILES	C3		ON OR	95 F9		VAL	EE	94
FN FRE	EI	8F	OCT\$	FF	99	VIEW	FF	D1
FIX	FF	9F	OPTION	8.8		VARPTR	EA	
FP05	FF.	A6	OFF	EE		WIDTH	9E	
GOTO	8.9		PRINT	91 BE		WAIT	96	D2
GO TO	89		PUT	98		WHILE	AF	
GOSUB	BD BD		POLL	FF	DF	WEND	86	
HEXE	FF	9A	POS	FF	91	WRITE	B5	
HELP	D9		PEEK	FF	97	WBYTE	FF	DD
INPUT	85		PSET	CF De		XOR	PA	
1SET LEEE	FF	E8	PRESET	EE	D3		F3	
1RESET	FF	El E2	PAINT	Di	D0 :	-	F4	
1F	88	62	PEN	FF	D8	*	F5	
INSTR	E8		RETURN	8E		/	F6	
INT	FF	85	READ	87 8A		×	F7 FE	
INP IMP	FF	98	RESTORE	8C		1	E9	
INKEYS	EF		RBYTE	FF	DE	>	FØ	
KEY		DB	REM	8F		-	FI	
KILL	CS		RESUME	A6		<	F2	
KANJI	DB		RSET	C7				
LOCATE	D6		RIGHTS RND	FF	88			
LPRINT	98		KIND	11	DD			



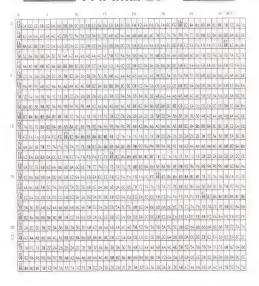
付録

- **80桁×25行モードVRAM番地表**
- ●40桁×25行モードVRAM番地表
- ●ROM内システムサブルーチン
- RAM上のシステムワークエリア
- ●システムI/Oポート
- Z-80(Z-80A)ニーモニック→機械語対照表
- ●10進数↔16進数変換表
- 1 バイト符号付16温数

|80桁×25行モード |VRAM番地表



'40桁×25行モード VPAM番地表



ROM内システムサブルーチン

番地	項目名	機能
0000	リセット	
8000	文法をチェックする	RST (またはCALL) の後に置かれているも
		ャラクタと(HL)と比較し、一致したらRST
		: 10Hに戻り,一致しなかったら Syntax error
		を表示する
0010	テキストから	テキストから1文字または数を読み込む
	1文字読み込む	
0018	1 文字出力	CRTまたはLPTに1文字出力を行う.Aレジ
		スタに文字コードを入れ RST 18H を実行す
		a
		(E64C) が 0 = CRT に出力
		0 ≠ LPT に出力
0038	ユーザ用 RST	モニタでは、ブレイクポイント用 or コマン
		ド待ちに戻るときに使う
036F	コマンド待ちへ移る	スタックを実行前に戻してコマンド待ちへ
		移る
047B	OK を出力する	OK を表示しコマンド待ちになる
04A7	コマンド待ち	コマンド待ちになる
05BD	リンクポインタセット	デキストのリンクポインタをセットする.
		NEWしたプログラムを復活するのに必要で
		ある
OB7C	RUN	RUNエントリ
009C	LET	LET文エントリ
18D9	LIST	LISTエントリ
3242	キースキャン	キースキャンを行う
		Cy=1:キーは押されていない

番地	項目名	機能
		Cy=0, Z=1:キーを押したまま
		Cy=0, Z=1:新しいキーを押した
		EEFEH にデータ
3583	キー入力	キーボード (キー入力用キュー) から1文
		字入力して Aレジスタに入れ、入力待ち
35C2	ブレイクのチェック	STOP, ^C のときキャリーフラグをセット
		して戻る
35CE	キー入力	キーバッファから 1 文字読む
		Z=1:バッファエンプティ
		Z=0: Aレジスタに入力データが代入されている
3E0D	CRT 出力	CRT へ A レジスタの 1 文字を出力する
		(Aレジスタ=ASCII コード)
3E9B	BELL	BEEP 出力
3EC0	BEEP ON/OFF	Aレジスタ=0:BEEP0
		Aレジスタキ⊪: BEEP1
3F79	ファンクションキーの	ファンクションキー([*1]~[*5])の表示
	表示	をする
		(E6B8) =0: 表示しない
		(E6B8) +0:表示する
3F7A	ファンクションキーの	ファンクションキーの表示
	表示	(E6B8) =0: 表示しない
		Aレジスタ=0: [*1]~ [*5]
		Aレジスタ=5: [f+6]~[f+10] を表示する
4021	■下行をクリア	ファンクションキーの表示を消し、テキス
		ト画面の種下行をクリアする
4047	タイマからデータを統	タイムバッファに時間のデータを読み込む
	み込む	FOODH←秒 (BCD ⊐一ド)
		F00EH←分 (BCD コード)

番地	項目名	機能
		FOOFH←時 (BCD コード)
		F010H←B (BCD ⊐ - F)
		F011H←月 (バイナリコード)
		F012H←年 (BCD コード)
428B	カーソルを消す	カーソルを表示しない
4290	カーソルを表示	カーソルを表示する
429D	VRAM のアドレスを求	キャラクタ座標 (H, L) が, どのアドレス
	める	に対応しているかを計算する
		レジスタ HL に与える座標の範囲が不適正な
		とき, "Illegal function call"が発生
4F01	NEW	NEW を実行する
4F21	CLEAR	変数の CLEAR を実行する
5550	文字列出力	HL レジスタの示すアドレスからデータ00h
		までの文字列を出力する
559A	ガベージコレクション	ガベージコレクションを実行する
5935	プリンタ出力	プリンタへ 1 文字出力する
5989	ブリンタ改行	LSOP + D (プリンタヘッドが行の先頭にな
		いとき) ブリンタを改行する
69FE	処理ルーチンのアドレ	中間言語 81H ~ D9Hの処理ルーチンアドレ
	ステーブル	ステーブル
6AB0	処理ルーチン〈FFシリ	中間言語 FF81~FFA9Hの関数の処理ルーデ
	ーズ前半) のアドレス	ンアドレステーブル
	テーブル	
6B02	処理ルーチン (FFシリ	中間言語 FFD0~FFE4H の関数・文としての
	ーズ後半)のアドレス	処理ルーチンアドレステーブル
	テーブル	1組4バイト
		前半 2 バイト: 関数の処理ルーチンアト
		レス

番地	項目名	機能
		後半2バイト:文の処理ルーチンアドレ
		2
6B56	予約譜インデックス	予約語テーブルの最初;文字による INDEX
6B8A	予約語テーブル	予約語と中間言語のリストが入っている
6E81	演算子テーブル	1 文字の予約語と中間言語のリスト
6E96	E3ROM 処理呼び出し	E3ROM 処理を呼び出すために使用
		1組4パイト
		CALL 4551H
		DB 〈処理#〉
		これを実行すれば、〈処理#〉に対する処理
		が実行される
6F6B	テキスト	テキスト画面のモード変更
	WIDTH	エントリ:8←桁数
		C←行側
		E6B2H スクロール開始行
		E6B3H スクロール終了行
		E6B4H ヌルアトリビュート
		E6B9H 00H: 白黒モード
		FFH:カラーモード
6FD1	CRTC セット	CRTC, DMACをセット
		エントリ:
		HLレジスタ=705BH(20行), 7066H(25行
		(E6C4, 5H) = VRAM 先頭 アドレス(F3C8H)
		(E6B9H) =00H(白黒), FFH(カラー)
		Cyフラグ = 1 (40桁), 0 (80桁)

『RAM上のシステムワークエリア』

番地	バイト数	機能・用途
E64B	1	LPOSの値(ブリンタヘッドの位置)
E64C	1	ブリンタフラグ (出力先の指定、0=CRT, 0=LPT)
E64D	1	LPRINT', ブリンタの横幅
E64E	1	WIDTH LPRINT で指定した値(プリンタの横幅を示す値が
		入っている)
E652	1	CTRL+0フラグ (画面出力をさせないためのもの)
E654	2	フリーエリアの上限値、CLEAR 文の第 2 パラメータ
E656	2	実行中の行番号
E658	2	テキストの開始アドレス(BASIC プログラム圏始番地)
E669		モニタ用ワークエリア
E690	2	RST 10H用ワーク (1EH, 10H)
E6A8	1	カーソル表示 ON/OFF コマンド
E6B0	1	テキスト画面ウィンドウ上限(スクロール開始行+1)
E6B1	1	テキスト画面ウィンドウ下隈(スクロール終了行)
E6B4	1	ヌルアトリビュートコード
E6B5	1	ヌルキャラクタコード
E6B6	1	CRT にコントロールコードを出力する
E687	1	未使用
E6B8	1	ファンクションキーを表示するフラグ
E6B9	1	テキストモード (OOH:白黒, FFH:カラー)
E6C0	1	ポート30Hへの出力データ
E6C1	1	ポート40Hへの出力データ
E6C2	1	ボート31Hへの出力データ
E603	1	ポート E4H への出力データ
E6C4	2	テキスト VRAM 先頭アドレス(初期値:F3C8H)
E6F2	16	ファンクションキー 「f・1」のデータ

番地	バイト数	機能・用途
E702	16	ファンクションキー f・2 のデータ
E712	16	ファンクションキー[f・3]のデータ
E722	16	ファンクションキー (・4 のデータ
E732	16	ファンクションキー[+5]のデータ
E742	16	ファンクションキー[f・6]のデータ
£752	16	ファンクションキーlf・7]のデータ
E762	16	ファンクションキー 1・8 のデータ
E772	16	ファンクションキー (・3 のデータ
E782	16	ファンクションキー[+10 のデータ
E826		モニタ用サブルーチン
EAC0	2	RST 10H用テキストポインタ
EAC2	1	RST 10H で読んだ文字
EAC3	1	RST 10H用 FAC のデータのタイプ
EAC4	8	RST 10H 用 FAC のデータ
EC27	1	Pオプションプログラムをセーブ中であることを示す
EC28	1	ドオプションブログラムをロード中であることを示す
EC29	1	Pオプション実行フラグ(0以外でLIST POKE MON SAVE
		がエラーとなる)
ECBB	1	フックアドレステーブル
EF85	1	1 行入力最後の桁
EF86	1	カーソルのY座標
EF87	1	カーソルのX座標
EF88	1	テキスト画面の行数
EF89	1	テキスト画面の桁数
EF90	1	ファンクションキー表示開始番号 (DOH:ノーマル, OS
		H:シフト
EF9A		テキスト画面行継続コード (00H:つながっている, 0A
		H: ^J, その他:つながっていない)

番地	バイト数	機能・用途
FOOD	Б	時間用バッファ
		FOODH←秒 (BCD □ - F)
		FOOEH←分 (BCD コード)
		FOOFH←時 (BCD コード)
		F010H←E (BCD □ - F)
		F011H←月 (バイナリコード)
		F012H←年 (BCD コード)
F01E	1	フォアグラウンドカラー
F01F	1	バックグラウンドカラー
F044	1	PAINT 文:領域色パレット番号、LINE文:色パレット番号
F21E	226	未使用
F320	168	未使用
FFF8	. 8	未使用

[■]未使用の部分は、ユーザが勝手に使用できるためよく使用される。ここを利用すると、テキストエリアを使わずに済む。

システム1/ロポート

MSB		内容	1/0	番地
BS2 BS1 MTON CDS ODLOR		システムコントロールポート	0	30
4D・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	LSB	MSB		
することによって画面の桁数が変えられ はないので達着すること。テキスト VRAA の読み出し方を雇える 0:40桁モード 1:30桁モード 1:30桁モード 1:カラーモード 1:白菓モード 1:白菓モード CDS・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	40	8S2 BS1 MTON CDS COLOR		
はないので注意すること。テキスト VRAA の読み出し方を雇える 0:40桁モード 1:30桁モード 1:30桁モード COLOR・・・・・・カラーモードか否かを設定 1:カラーモード 1:白黒モード CDS・・・・・・・カセットへ出力する信号の切り替え MTON・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	直接操作	4040桁モードか否かを設定。このビットを		
の読み出し方を置える 0:4B所モード 1:80所モード 1:80所モード 1:10所モード COLOR・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	しるわけて	することによって画面の桁数が変えられ		
○:40桁モード 1:30桁モード 1:30桁モード COLOR・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	4M の内容	はないので注意すること、テキスト VRAI		
1:80桁モード COLOR・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・		の読み出し方を置える		
COLOR・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・		0 :40桁モード		
#:カラーモード 1:白菓モード 1:白菓モード CDS・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・		1:80桁モード		
1:白黒モード CDS・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・		COLORカラーモードか否かを設定		
CDS・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・		Ⅱ:カラーモード		
MTON		1:白黒モード		
BS1, BS2 ·····USARTをカセットインターフェースに使 RS-232C に使うかを選択		CDS ······カセットへ出力する信号の切り替え		
RS-232Cに使うかを選択		MTONカセットのモータの ON(1), OFF(0)		
	きうか。			
90		RS-232C に使うかを選択		
30 1 モートセレクト		モードセレクト	1	30
MSB	LSB	MSB		
SW1 SW1	SW4 S1			
ディップスイッチの値を読み込む		ディップスイッチの値を読み込む		
II : ON		II : ON		
1 : OFF		1 : OFF		

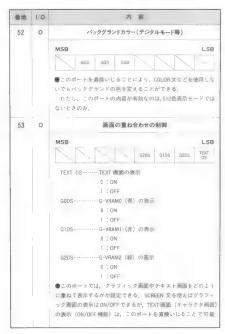
番地	1/0	内 客
31	0	グラフィックコントロール
		MSB LSB
		25 LINE HOOLOR GRPH RMODE MMODE LINE
		200LINE 専用高解像ディスプレイモードのコントロール
		0 : 400Line
		1 ; 200Line
		MMODERAM モードコントロール
		0 : ROM, RAM € F
		1 : ALL RAM ₹ — F (64K RAM)
		このビットが∥のときは一切のROMが禁止される。
		このモードは CP/M などのアプリケーションを走
		らせる場合に使用する
		RMODE ·······ROM モードコントロール
		D : Nag-BASIC
		1 : N-BASIC
		選択される ROM のバンク, テキストウィンドウの
		可不可などを選択する
		GRPH・グラフィックディスプレイモード
		0 : 表示禁止
		1 : 表示
		HCOLORカラーグラフィックディスプレイモード
		0:白黑
		1:カラー
		単色3ページモードか、カラー1ページモードか
		の選択をする
		25LINE専用高解像ディスプレイ LINE コントロール
		□ : 20LINE モード
		1 : 25LINE

番地	LO		内	容		
31	1		±−K-	セレクト		
		MSB				LSE
		SW4 SW3 S2 S0	SW2 SW2 S6 S5	SW2 SW2 S4 S3	SW2 S2	SW2 S1
		ディップスイ	ッチの内容を取り	7込む		
		0 : ON				
		1 COFF				
32	1/0		モート	指定		
		MSB				LSE
		SINTM GVAM	PMODE . TMODE	AVC2 AVC1	EROM SL1	EROM SL0
		EROMSL1	EROMSLO	····N ₈₉ -BASIC 7	·拡張 RO	Mの番号
				を選ぶとき	こ使用す	る
		0	0	: 内部 EROM	ベンクⅡ	を選択
		0	1	: 内部 EROM	ベンク 1	を選択
		1	0	: 内部 EROM	ベンク 2	を選択
		1	1	: 内部 EROM	ベンク 3	を選択
		AVC2	AVC1	·····AV コントロ	ール	
		0	0	: テレビ/ビ	デオモー	F
		0	1			
		1	0	: アナログ RG	B E-F	
		1	1	:オプション	€-ド	
		TMODE	…高速 RAM 選択			
			』:高速 RAM			
			1:メインRA	M		
			高速モード時に	こはメインRAM (7) F000 ~	FFFFH看
			地までが高速F	RAMにも切り替	えられる	. 859
			の RAM を選択	するか決めるビ	ット	

番地	1/0	内 容
32	I/O	PMODE ··········カラーモード
		0:ディジタルモード
		1:アナログモード
		カラーパレットの内容がディジタル RGB か、ア
		ナログRGBかを決める
		GVAM ·······G-VRAM アクセスモード
		D:独立アクセスモード
		1:拡張アクセスモード
		PC-8801mkIISRで採用されたALUによる画面ア
		クセスモードか、従来どおりの3回に分けてア
		クセスするモードかを決める(ALU:Arithme-
		tic Logic Unit)
		SINTM ·····PSG 割り込みマスク
		■:割り込み許可
		1:割り込み禁止
		PSG を使ってバックグランドで音を出すか出さ
		ないかの制御
34	0	G-VRAM制御
		MSB LSB
		ALU ALU ALU ALU ALU ALU ALU 00
		0:ALUn1, 0:ALUn0ピットリセット
		O:ALUn1、I:ALUn0······ピットセット
		1:ALUn1, 0:ALUn0ピット反転
		1:ALUn1, 1:ALUn0ノーオペレーション
		(n = 0 ~ 2)
		●拡張 G-VRAM アクセスモードで各ページの ALU にどのような演
		算をさせるかを指定する。

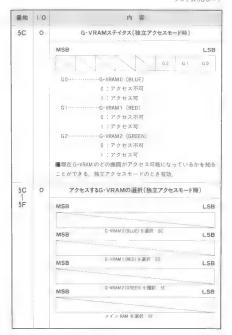
番地	1/0	内 客
35	0	G-VRAM制御
		MSB LSB
		GAM GDM1 GDM0 PLN2 PLN1 PLN0
		PLNO···········G-VRAMOの比較データ
		PLN1G-VRAM1の比較データ
		PLN2G-VRAM2の比較データ
		GDM1 GDM0 …グラフィックデータマルチブレクサ¶御
		 0 …ALUの出力を同時ライト
		0 1 …リードデータを同時ライト
		1 0 ···G-VRAM1→G-VRAM0 仁転送
		1 …G-VRAM0→G-VRAM1 に転送
		GAM······G-VRAM アクセスモード
		□:メイン RAM
		1 : G-VRAM
		■拡張 G-VRAM アクセスモードで、どこからのデータを移すかと
		いったことを制御する。
40	0	ストローブボート
		MSB
		FBEEP JOP1 GHSM CLDS CCK CSTB PSTB
		PSTBブリンタストローブ
		出力ポートIDHにセットされているブリンタへ
		の出力データを出力するときに、このビットを
		1→0→1と変化させる
		CSTBカレンダクロックストローブ
		カレンダ時計にコマンドやデータを出力すると
		き、10Hポートにデータをセットし、このビッ

番地	1/0	内 容
40	0	CCKカレンダシフト用クロック
		CLDSCRT インターフェース同期コントロール
		GHSM···········グラフィックスハイスピードモード
		il: 高速モード
		1:標準モード
		BEEPビープ音コントロール
		0 : BEEP OFF
		1 : BEEP ON
		このビットを上げ下げすることによって、内蔵
		ブザー(カセットインターフェースのキャリア
		の音)を ON/OFF できる
		JOP1汎用出力ポート
		FBEEP・・・・・・・サウンド出力
40	I	各種データ
		MSB LSB
		VRTC CDI SW2 DCD SHG BUSY
		BUSYブリンタ BUSY
		Ⅱ:プリンタ READY
		1: プリンタ BUSY
		現在プリンタが、データが受信できる状態かど
		うかを知ることができる
		SHG高解像度 CRT モード
		[]:高解像
		11 · [60] [79] [8]
		1:/ーマル
		1:/ーマル
		1 : /ーマル DCD



番地	1/0	內 容	
53	0	になる.ただし,TEXT画面がON/OFFできるのはカラーックモード時のみ,	グラフィ
54	0	カラーパレットの制御(デジタルモード書)	
5B		MSB	LSB
		PG0 PR0 PR0 PR0 PR0 PR0 PR0 PR0 PR0 PR0 PR	PB0
		パレット番号1の色 55	P81
		バレット基号2の色 56 PG2 PR2	PB2
		パレット番号3の色 57	PB3
		パレット番号4の色 58	PB4
		PG5 PR5 PR5 PR5 PR5 PR5 PR5 PR5 PR5 PR5 PR	PB5
		パレット書号6の色 5A	PB6
		パレット書号7の色 5B	PB7
		PGn・・・・・・パレット n 書の GREEN	
		PRnパレット n 番の RED PBnパレット n 番の BLUE	

番地	1/0				内	容			
54 5 5B	0	●BASIC の 割り付け を使用す	を決め	るボート	. P+1	コグ時も	同じ54日		- , ,
54	0		カラ	ラーバレッ	トの制御	即(アナロ	グモート	(
5B		MSB							LSB
		0	9	PR02	PROI	PROO	PB02	PB01	PB00
		バレット書	号目の	※、青の亨	るさのレ	ベル (回	同)を設定		
		0	1	1		1	PG02	PG01	PG00
				…パレッ	F n ≣	の冊の階	調		
£.4	0	PGn2~ ■アナロ き赤。 ■ になる。	PGn 0- グモー の階調	…バレッ ド時には , 第6と	ト n 器(は, これ) ごットが	らのポー 1 のとき	調 トは第6 調の階類	層を指定	すること
54	0	PGn2~ ■アナロ き赤。 ■ になる。	PGn 0- グモー の階調	…バレッ ド時には	ト n 器(は, これ) ごットが	か緑の階 らのボー 1 のとき	調 トは第6 調の階類	層を指定	すること
54	0	PGn2~ ■アナロ・ き赤。電 になる。	PGn 0- グモー の階調	…バレッ ド時には , 第6と	ト n 器(は, これ) ごットが	か緑の階 らのボー 1 のとき	調 トは第6	ード酬)	すること
54	0	PGn2~ ■アナロ・ き赤。 になる。 MSB	PGn 0- グモー の階調 パック	…バレッド時には ,第6と グランドフ	ト n 器 c は, これ・ (ットが カラーの)	か緑の階 らのボー 1 のとき	調 トは第6 関の階類 ナログモ	ード酬)	すること LSB
54	0	PGn2~ ●アナロ・ き赤、 ■ になる。 MSB 1 BGB2~ BGR2~	PGn 0- グモー の階調 パック 0 1 - BGB 0	…バレッド時には ,第6と グランドフ	ト n 番 c c c c c c c c c c c c c c c c c c	か緑の階 らのボー 1 のとき	トは第6Mの階割ナログモBGB2	を指定 -ド聯 BG81	すること LSB BGBO



番地	1/0	内 窖											
5C { 5F	0	●これらのポートは、独立アクセスモード時にどの G-VRAM を クセスするかを選択する。それぞれ出力するデータに関係なく、 そのポートへ何らかのデータを出力することによって対応するG VRAMが選択される。											
70	1/0	オフセットアドレス											
		MSB LSE											
		AP15 AP14 AP13 AP12 AP11 AP10 AP9 AP8											
71	0	出せば、現在テキストウィンドウに映されているのがメモリ空間											
71	0	出せば、現在テキストウィンドウに映されているのがメモリ空間 のどこかを知ることもできる。 N ₈₈ -BASIC拡張ROM制御											
71	0	N _{ee} -BASIC拡張ROM制御											
71	0	出せば、現在テキストウィンドウに映されているのがメモリ空間 のどこかを知ることもできる。 N ₈₈ -BASIC拡張ROM制御 MSB											
71	0	出せば、現在テキストウィンドウに映されているのがメモリ空間のどこかを知ることもできる。 Note: BASIC拡張ROM制御 MSB LSE EROM7 EROM5 EROM5 EROM6 EROM5 EROM2 EROM1 VERIOM EROM1 — EROM7 0 : 道訳 1 : 道訳しない IEROM											
- 71	0	出せば、現在テキストウィンドウに映されているのがメモリ空間のどこかを知ることもできる。 N ₈₀ -BASIC拡張ROM制御 MSB											
- 71	0	出せば、現在テキストウィンドウに映されているのがメモリ空間のどこかを知ることもできる。 N ₈₄ - BASIC拡張ROM制御 MSB LSE EROM1 EROM8 EROM8 EROM8 EROM2 EROM1 REROM EROM1 - EROM7 - 0 : 道釈 1 : 道択しない IEROM - 内部拡張 ROM (EROM — €3ROM) 遊択 0 : 道訳 1 : 道択しない ◆メモリ空間の5000H - 7FFFH 番間の空間に割り当てられたバンクROMの遊訳をする。当然、一度に複数のピットを 0 にすることは											
71	0	出せば、現在テキストウィンドウに映されているのがメモリ空間のどこかを知ることもできる。 Notes BASIC拡張ROM制御 MSB LSE ERDM7 ERDMS ERDMS ERDMS ERDMS ERDM2 ERDM3 ERDM2 ERDM1 IERDM EROM1 — EROM7 0 : 選択 1 : 選択しない IERDM IERDM											



*キーボードのとり込みに関しては本文第3事参照のこと。

8ピット・ロード

						5	3		A	В	ĺ		D		1			H				HZ.	BC)		281	12	X d)			Çne		8
	1				(2) (2)	E 5	D	2	F	7.8		0	7.5	i	7	B	1		7			7 E	οA	1	A		E		PE	30.0	A	3 8
LD	E							4	7	1.0	4	t	13		4	3	,		4	5		1 6					6		16			0 E
LĐ								4	F		4										Ī,	3				E	D	ļ				O F
LD	ı							5	7	5.0	5																5		0 6			1 8
Lb	2							5	F	5 8		9	5/									s E.				O LL	Đ		108			1 5
LD	î							6		5 (6			Ì		3	6	. 4	6	5		5 11					D					2 6
Lb	L							6	F	5 8	Б	9	6.8		Б	В	. 6		6	Eı		3.8				5	DE	E	3			2 1
		ıL								7.0		1	7.5		7				7							Second						3 6
																																n
LD		E						1																								
D	13		ľ.					d	7				Ä				E ST	A														TO DAY
D I	IŸ								7		Erro																					F 3 6
LD	10	n I						300	24																							
tt	L							E 4	U 7																							
LD	ŀ							E	DE																							

16K > h - p - K

		- In - I								
		AF	BC	DE	HL	SP	IX	[Y	nn	883
LD	AF.									
	BC.								0 1	E D 4 B
LD	DE,		+						11	E D
LD.	HL.								21	2.4
LD	SP,				F9		DD F9	F D F g	31	E D

プロックを

LDI	ED A 0
LDIR	ED B0
LDD	ED A8
LDDR	ED B8

LD 18.								21	2 A
I.D. IV.								F 1	FD 2A
LD nn		ED3	E D 3	2 2	ED 73	DD2	F D 2 2 2 2 2 2		. 1
	F5	C 5	D 5	E 5		DD E5	E5		
Por -	Fi	C 1	DI			DD	FD E1		

ブロック・・	サーチ
CPI	ED A 1
CPIR	ED ED
CPD	ED A 9
CPDR	E0 89

TEN STABLES

		Α	В	C	D	ε	В	1.	BLI	(IX)	TY do	n
ADD	Α, -	8.7		8.1		83				D()	FD BB	0.0
ATH:	Λ	8.5				8 B				SE	SE.	CE
SUR		9.7		9.1		9.3				D D 9 6	FD 96	D e
SHC	A					9.B	9 C	9 D		DP	FD 9E	08
AND		A 7	A ()		A 2		A4		A 6	DD A	FD A6	E
- XOR		AF	A 8	A 9	ΑΑ	AB	AC.	A D	AΕ	PP	FD AE	E
R		B 7	Ве	H-)	B 2	В 3	B 4		6.6	DD B 6	FD B 6	100 0
CP -		BF	B.R	R 9	ВА	88	вс	ВР	BE	DD	FD.	FI
INC.		3(0.4	0.0	14	10	2.4	2 °C	3.4	DD 34	FD 34	
DEC		3.17	0.5	0 D	15	10	2.5	2 D	3.5	DD 3.5	FD 35	

CPUD>FD-A

NOP	0.0
HALT	7.6
DI	F
EL	FF
0 M1	E1
IM I	E 5 4
194 2	E 5

16ビット算術演算

	BC	DE	BL :	SP	1X	IY
ADD BL. ×	0.9	19	29	3.9		
ABD IX.	0.9	pp 19		DD 3 9	DD 29	
ADD IV.	FD 09	FD 19		FD 3.9		FD 29
Abc III.	ED	ED	EΠ	ED 7A		
SBC HL	ED 42	E D 5 2	ED	ED		
INC ·	03	13	23	33	DD 23	F D 2 3
DEC ×	0 B	18	2 B	3 B	D D 2 B	FD

TA 7 4.

EX	AF. AF	0.8
EX	DE. HL	EB
EX	SP , HL	E 3
EX	SP.IX	E 3
EX	SP , IY	FB E3
EXX		1) 9

アキュムレータ操作

DAG	2.7
CPL.	2 F
NEG	ED
NEG	4.4
CCF	3 F
SCF	3.7

	. A	В	С	D	E	В	L	HL	11X + d	· IY
RLC -	10 B 0 7	. 0 0	0 B	CB 02	CH 03	CB 04	C H 0 5	CB 0.6	11 to	18
RRC +	CB gF	0.8	0.9 C.B	CB 0A	CB 0B	⊕B ⊕C	CB 0D	0 E	100	18
RL .	CB 17	CB 10	11 11	12	CB 13	CH 14	CB 1.5	CB	80	1000
RR .	CB 1F	CB 1.8	6 B	CB LA	18	CB 1C	CB LD	CB 1 E	UR TE	7.00
SLA	CB 27	2 0	2 I	CB 22	CB 23	C B 2 4	C B 2 5	CB 2.5	28	SECTION OF
SRA ×	CB 2F	CB 28	C B 2 9	CB ZA	CB 2B	CB 2C	CB 2D	CB 2E	DON H	Santa N and
SRL .	CB 3F	CB 3.8	3 9	CH 3A	CB 3B	CB 3C	CB 3D	C.B 3 E	88	20 K
RLD								ED 5F		
RRD								ED 67		

	1	A
RLCA		0.7
RHCA		0 F
Rt.A		1.7
RRA		1 F

	COND	С	NC	2	NZ	PE	PO	М	Р	
JP ×. nn	C 3	DA h	() 2 n	CA 1	U 2	ΕÁ	E 2	FA E	F2	
JR ×, r	18	38 e 2	3.0		2 D r 2					
JP (HL)	E 9									
JP (IX)	DD E9									
JP 'IV'	FD E9									
CALL ×, se	c D	D-C	D4	CC o	C 4	EC n	E 4	FC	F4	
DJNZ +										10
RET .	C 9	Вd	D 0	CB	Ce	Ев	Εo	Fs	F 0	
RETI	ED 4D									-
RETN	ED 45									

RST	Hoo	C 2
RST	O&H	CF
RST	1011	D.7
RST	18H	DF
RST	2016	E 7
RST	28 H	EF
RST	30 H	F 7
ast	38 H	FF

E	7	M	я	fr.

		A	В	С	D	E	Н	L	(RL)	-H	HY
BIT	0, ×	CB 47	CB 40	CB 41	C8 42	CB 43	CB 4.4	CB 4.5	CB 4-5	DOS 6	ECG 6
BIT	1. ×	CB 4F	CB 4.5	CB 4.9	CB 4A	CB 48	CB 4C	0B 4 D	CB 4E	DD 4 4 6 E	CH
BIT	2, ×	CB 5.7	CB S 0	CB 51	C8 52	CB 53	CB 5.4	CB 55	CB 56	OB CB	000
BIT	3. ×	CB SF	CB 58	CB 5 9	CB 5A	CB SB	CB 5C	CB SB	CB 5E	DB E	E SEC
BIT	4 , ×	CB 67	CB 60	CB 61	C B 5 2	CB 53	CB 64	CB 65	C B 5 6	000	EC-8
BIT	5, ×	CB 6F	68 68	6 B	CB SA	GB 6 B	CB	CB 6D	CB SE	DO THE	E L
BIT	6, ×	CB 77	CB 70	CB 71	CB 7.2	CB 73	CB 74	CB 7.5	CB 76	DD CB d 7.6	E H
718	7. ×	CB 7F	CB 78	CB 79	CB 7A	CB 7B	CB 7C	CB 7D	CB 7E	CB 4 7 E	CB d TA
RES	0, ×	CB 87	C8 80	CB 81	CB 82	CB 83	C B 8 4	CB 85	CB 86	G a a	S B B
RES	1. ×	CR 8F	CB 88	C B 8 9	CB 8A	CB 8B	CB 8C	CB 8D	8 E	00 E	20 E
RES	2, ×	CB 9.7	CB 9 0	CB 91	CB 9 2	CB 93	CB 94	CB 95	CB 96	DG 6	EC-ee
RES	3, ×	CB 9F	CB 98	CB 99	CB 9 A	6 B	CB 9C	CB SD	CB 9E	C. a. e.	EC.Da
RES	4. <	CB A7	CR A 0	CB A1	CB A2	CB A3	CB A4	CB As	CB A 5	CB CB	PDH A 6
RES	5	CB AF	C B A B	CB A 9	CB AA	C B A B	CB AC	CB AD	C B A E	Ça A E	E B
RES	6. ×	CB B7	CB Bo	CB B1	CB B2	CB B3	CB B4	CB Bs	CB B6	DB GB GB	E 8
RES	7	BF	CB Bs	CB B9	CB BA	CB BB	C B B C	CB BD	CB BE	DD CB BE	SCH E
SET	0. 1	CB C7	CB Co	CB C1	CB C2	CB C3	CB C4	CB C5	CB C6	PB PB	SAC S
SET	1.7	C B C F	CB C8	CB C9	C B	CB CB	CC	CB	CB CE	D C T C	0.00 E
SET	2. %	CB D7	CB D0	CB D1	CB D2	C8 D3	CB D4	CB D5	CB D6	D D D	E D B
SET	3, ×	CB DF	CB D8	CB D9	CB DA	CB DB	CB DC	CB	CB DE	0 8 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	200
SET	4. %	CB E7	CB E 0	CB E1	CB E2	CB E3	CB E4	CB Es	CB Es	Dawner Parent	EC. 8
SET	8, 1	CB EF	CB E8	CB EB	CB EA	CB EB	CB EC	CB ED	CB EE	E.E.	ECAN E
SET	6	CB F7	CB F0	F1	CB F2	CB F3	CB F4	CB F5	CB F6	DB CB	EC. al
SET	7. <	CH FF	CB F8	CB F9	CB FA	CB FB	CB FC	CB FD	CB FE	DU SE	140.0 140.0

λ	20	
IN	A , n	DE
ÍN	A, C)	E E
IN	B, (C)	E E
IN	C, (C)	E E
EN	D _i (C)	5 C
BN	E. (C)	5 8
IN	$H_{\varepsilon} \circ C :$	E 0
IN	L _i (C)	6.8 E.C.
LNI		ED A 2
101	R	EU B 2
INT		ED
INE	ıR	EDBA

出力

1000	** 0		
CKIT	ic.	A	ED 79
OUT	· C.,	В	ED 41
OUT	: C),	С	ED 49
OUT	e,	D	ED 51
OUT	· C ·.	E	ED 59
OUT	(C.	Н	E D 6 1
OUT	С,	L	ED 69
OUTI			E D A 3
OTIR			E D
OUTD			E D
OTDR			ED BB

引用資料: NEC技術資料 #COM-82インストラクション活 用表(IEM-656A JAN-12-80)

L	15	33	47	63	79	96	111	127	143	159	175	191	202	223	239	255
ш	14	30	46	62	38	96	110	126	142	158	174	190	206	222	238	254
0	13	58	45	19	11	83	109	125	141	157	173	189	205	221	237	253
O	12	28	44	09	16	92	108	124	140	156	172	188	204	220	236	252
00	=	120	43	23	75	16	107	123	139	125	171	187	203	219	235	251
<	10	26	42	288	74	90	106	122	138	154	170	186	202	218	234	250
on	6	25	41	23	73	89	105	121	137	153	169	185	201	217	233	249
00	00	24	40	29	72	00	104	120	136	152	168	184	200	216	232	248
r-	7	23	39	25	71	00	103	119	135	151	167	183	188	215	231	247
9	9	22	38	54	20	98	102	118	134	150	166	182	198	214	230	246
us.	ın	21	37	53	69	50	101	13.7	133	149	165	181	197	213	529	245
~2	4	50	38	52	. 89	8.4	100	116	132	148	164	180	196	212	228	244
~	· en	19	35	10	19	60	56	115	131	147	163	179	195	211	227	243
~	2	00	34	50	99	82	98	114	130	146	162	178	194	210	526	242
-	-	17	33	49	99	00	97	113	129	145	161	177	193	209	225	241
0	0	16	32	00	64	80	96	1112	128	144	160	176	192	208	224	240
南山田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田	0	-	2	1 17	4	120	0	7	00	on .	V	B	O	0	ω	ш

E 49	0.0	7	2	m	4	ιΩ	φ	7	00	6	¥	c 0	O	Q	ш	LL
-	0	-	2	69	42	5	100	-	00	10	10	=	12	13	14	15
	16	17	00	9	20	21	22	23	24	25	56	27	28	59	30	31
	32	33	34	35	36	37	39	33	40	41	42	43	44	45	46	47
	48	49	50	51	52	53	5.4	55	99	22	10	29	9	19	62	63
	64	99	99	67	68	69	70	7.1	72	73	74	75	16	77	78	79
	80	18	82	633	84	85	98	00	90	89	8	91	36	93	94	92
	96	97	98	66	100	101	102	103	104	105	106	107	108	109	110	E
	112	113	=	115	116	1117	118	119	120	121	122	123	124	125	126	127
	-128	-127	- 126	-125	-124	-123	-122	-121	- 120	-119	- 118	-117	-116	-115	-114	-113
-	-112	E -	- 110	- 109	- 108	-107	- 106	- 105	-104	- 103	-102	-101	- 100	- 99	1 98	- 97
	98-	- 95	-94	-93	-92	- 91	- 90	- 89	1 000	-87	-86	- 85	- 84	- 83	- 82	-8
	- 80	- 79	- 78	-77	- 76	- 75	- 74	- 73	-72	12	-70	- 69	- 68	-67	99-	-65
	-64	-63	-62	19-	09-	-59	100	-57	1.56	- 55	-54	- 53	- 52	-51	-50	- 49
	- 48	-47	-46	-45	-44	-43	- 42	-41	- 40	-39	-38	-37	-36	-35	-34	- 33
	- 32	-33	-30	-29	- 28	-27	- 26	- 25	-24	-23	-22	-21	-20	- 19	90	-17
	116	10	- 14	0	0	111	9			1	L		4			

参考文献

- ●「PC 8801mk[ISRユーザーズマニュアル」 NEC
- ●「PC-Technow8800mkII」 平松、 八木 共著 システムソフト社
- ●「PC 8801mktiSRテクニカルマップ! ソフトウェアを知る」小林、今野、平間 共著 券和システムトレーディング(株)
- ●「電子科学」 1983年 1月号 産報出版
- Oh PC」1983年 11月号。1984年 3月号 日本ソフトバンク社

索引

数字	のリセット133、142
16進数17	RET命令······ 245
と10進数の対応330	DEC命令
の表し /j18	DI命令······ 163, 242
€ - 1 - 40	DMA228, 252
(符号付)331	DUAD 88D 35, 278
2 遊数 15	D コマンド 41
と10進数の対応 15	EI命令163. 242
の表し方16	EXX命令243
- を16進数に直す 19	EX命令243
3 原色による色69	E コマンド42
40桁×25行モードVRAM番地表306	F = 7 > k43
6800系CPU20	F レジスタ······ 25
8080A と Z-80(Z-80A) CPUの	の各ピットの意味 26
命合对応表34	G-VRAM操作士の注意163
8080 ACPU20	G コマンド 44
8 進数モード 40	HELPコマンド 54
80桁×25行モードVRAM番地表 304	1/0アドレスマップ29
	I/Oボート······ 46, 49, 113, 286, 314
アルファベット	I(割り込みベクトル)······22
AND命令121	[FF(インタラブト・レジスタ) 22
ASCII ⊐ - F 72	INC命令81
A コマンド 39	IN命令······113, 254
BASICインタブリタ・・・・・・ 292	(Cレジスタ間接)····· 113
B ⇒ ⇒ > F · · · · · · · · 40	[X, [Y(インデックス・レジスタ) 22
CALL命令 ······· 245, 247	I コマンド46
CPUによるアセンブリ言語の違いと	JP命合82
Jt;i6i.4:	LD命令····· 80, 246
CRTC252	LDDR命令281
CRTのカラーの原理69	LDD命令280
コントローラ270	LDIR命令97, 268, 280
CTRL-B=>> F 55	LDI命令······279, 282
CTRL-D=> F 56	L コマンド
CTRL-R 2 >> 57	M コマンド・・・・・・・・・・・48
CTRL-W=マンド 58	NEW命令262
CYレジスタ197	OP⊐ - F · · · · · 23
Cyフラグ26、197	OR命令124

OUT命令 114、254 ア・バライン 70、88 - (Cレジスタ間後) 115 ア・ドリビュートを到情 110 ファンド 49 ア・リビュートを到情 141		
ロコマンド・ 49	OUT命令114, 254	ア・バライン 70, 88
PCIT ログラム・カワンタ) 22 アナログRCB 244 PUN音合 247 アナログモート 284 PUN音合 247 アナウグモート 284 PUN音合 247 アナウブモート 284 PUN音合 247 アナウブモート 284 PUN音合 247 アナウブモート 284 PUN音合 247 アナウブモート 248 PUN音合 247 PUN音の音 247 PUN STAN STAN STAN STAN STAN STAN STAN STA	(Cレジスタ間接) · · · · · 115	アトリビュートの属性141
POP命令	O コマンド49	アトリビュートエリア 70
PUNIA 12	PC(フログラム・カウンタ) ·········· 22	7 + 12 7 RGB 284
ドナフンド・・ 49	POP 命令247	アナログモード・・・・・・・・・・284
P コマンド・ 49 ウィンドウ温砂・ 255 R(ノモリ・リファンニ・シスタ) 22 エント リポイント 272 RET 命令 247 オーペーラップ・ 882 オープ・エンド・ 198 オープ・エンド・ 556 オープ・エンド・ 568 オープ・エンド・ 568 オープ・エンド・ 568 オープ・エンド・ 568 オープ・エンド・ 568 オープ・エンド・ 569 オープ・エンド・ 569 カーディン・ド・ 569 カーボード・ 569 カーボード・ 569 カーボード・ 569 カーボード・ 569 カーボード・ 569 カーズ・エンド・	PUSH命令247	アンダライン
P コマンド・ 49 ウィンドウ温砂・ 255 R(ノモリ・リファンニ・シスタ) 22 エント リポイント 272 RET 命令 247 オーペーラップ・ 882 オープ・エンド・ 198 オープ・エンド・ 556 オープ・エンド・ 568 オープ・エンド・ 568 オープ・エンド・ 568 オープ・エンド・ 568 オープ・エンド・ 568 オープ・エンド・ 569 オープ・エンド・ 569 カーディン・ド・ 569 カーボード・ 569 カーボード・ 569 カーボード・ 569 カーボード・ 569 カーボード・ 569 カーズ・エンド・	ドオブション解除272, 275	インデックス・レジスタ138
R(イモリップアレッシュレジスタ) 22 エントリポイント 272 RET命令 247 RL 198 オフセットアレスレジスタ 256 RAM ! のシステムワークエリア 311 オペランド 64 ROM ! のシステムワークエリア 198 BP(スタック・ホインタ) 22 BP R ファンド 50 カラーボレーシの別り接 163 カラ・モンド 70 アラブィックス・ドット構成 159 開催コントロール 96 S ファンド 50 オーボード・デ・タ 117 オンアンド 50 オーボード・デ・タ 117 YRAM 162 キャラクタ・コード 73 Z1 トッションド 52 RE REM 96 PR		
RET 命令		エントリポイント・・・・・・・・272
RL 98 オアセットアドスレンスタ 256 RAM トのシステムワークエリア 311 オペランド 64 ROMPリーチシー 106. 307 RR 98 オペランド 50 カウンタ 205 R3マンド 50 カウンタ 205 オウンタ・205 R3マンド 50 カウンタ・205 オウンタ・205 オウンステンタ・205 オウンステンステンステンステンステンステンステンステンステンステンステンステンステン		オーバーラップ・・・・・・・・・・・・282
RAM ドランステムフース・リア 311		
ROMはルーチン 106、307 RR R 198 RST命令 249 クランタ 205 R フマンド 50 クラーバレット 255 SLA 98 タアス・ホインタ 198 SP(ステック・ホインタ) 22 クラーボレット 163 SP(ステック・ホインタ) 24 クラーボレット 169 のグライクストドン構成 159 のグライクストドン構成 159 のグライクストドン構成 159 のグライクストドン構成 159 ルス・ロマップ 28 S コマンド 50 キーボード ア・タ 117 オフマンド 51 キーア・ア・タ 117 VRAM 162 キーファイン 240 ソコマンド 52 縁後 250 VJマンド 52 縁後 264 VJマンド 52 縁後 264 XJマンド 52 縁後 264 XJマンド 52 縁後 652 XOR命令 128 ボアと・ブラ 47 Xファンド 52 縁後 652 XOR命令 128 ボアと・ブラ 47 Xファンド 52 縁後 652 XOR命を 128 ボアと・ブラ 47 Xファンド 52 縁後 652 XOR命を 128 ボアン・ブラ 47 Xファンド 52 ボアン・ブラ 47 Xファンド 52 ボアン・ブラ 47 Xファンド 52 ボアン・ブラ 47 Xファンド 52 ボアン・ブラ 27 アン・ブラ 24 35 ドン・ブラブ 27 アン・ブ・ブルボ 59 シークレット 70、856 カイナ 59 シークレット 70、856 カイナ 59 シークレット 70、856 カイナ 70、556 カイナ 70・70・70・70・70・70・70・70・70・70・70・70・70・7	RAM トのシステムワークエリア 311	
RR 198 カゲ行 249 カウェー 255 R コマンド 50 カラー・ベレート 285 SLA 198 カラー・ブレーンの切り付え 63 SP(スティク・ホインク) 22 ファー・ドー・クラフィクス・ド・特別 151 Mボーント 587 J マンド 50 キーボード・デ・タ 117 アステー 50 キーボード 73 21 カー・で観り立り上が 52 接近の 52 マンド 53 マンド 54 ボード 72 スピーク 73 21 カール 52 を行 73 スピーク 74 スピーク 75 スピーク		
RST命令 249 カウンク 205 RJマンド 50 カラーゼレト 285 SLA 198 SP(ステック・ポインク) 22 カラーゼレーンの切り替え 163 SP(ステック・ポインク) 22 カラーゼレーンの切り替え 163 SP(ステック・ポインク) 22 カラーゼレーンの切り替え 163 SP(ステンド 198 RSPG-VKAM 162 端菜ドット) 59 のメモリマップ 28 間様ドット) 59 のメモリマップ 28 間様アドット 29 VRAM 652 キャブクタ コード 73 271 ハの移動 260 キャブクタ コード 73 271 カステムア 27 27 27 27 27 27 27 27 27 27 27 27 27	RR 198	か行
R コマンド 50 カラーボレット 285 SLA 186 SP(スタック・ホ (ンタ) 22 SR: 98		
SPICスタック・ボインタ) 22 カラ・モード 70 NR	R ==> F 50	
SPIC タ・ク・ボインタ) 22 カラ・モード 70 70 70 70 70 70 70 70 70 70 70 70 70	S1.A198	カラーブレーンの切り替え163
SRI. 98	SP(スタック・ホインタ)22	カラーモード・・・・・・・・・・・・ 70
- のグラフ (マクス・ドン構成 159 両値コントロール 96 96 5 コンド・ 50 キーボード・デック 117 177 1		時のG-VRAMの番地161
- のメモリマップ 28 開発 ドレッシング 240 S ファンド・ 50 キーボード・ア・タ 117 TM コマンド・ 51 キーブード・ 292 VRAM・ 162 キーワック・コード・ 292 VRAM・ 260 キーワック・コード・ 73. 71 ションド・ 52 機関金 652 Wコマンド・ 52 機関金 652 Wコマンド・ 52 ボリコンド・ 52 ボリコンド・ 52 ボリコンド・ 52 ボアセンブラ・ 47 ステーズー・ 167 スキーフード・ 20 WW アラグ 27 ファンド・ 54 サーフェス・ 64 アスキーフード・ 294 サーフェス・ 56 テンキ・アドレス 249 サーフェス・ 56 アセンブラ・ 24 35 ドスタート・アドレス 249 ファンド・ 59 シークレット・ 70. 86 フリボー・ 59 シークレット・ 70. 86 フリドー・ 70. 86 フリボー・ 59 シークレット・ 70. 86 フリボー・ 70. 86 フリズー・ 70. 86 フ	SRØG-VRAM162	商素(ドット)159
5 コマンド 50 オーボード・ア・タ 117 TM コマンド 51 オーラード 192 VRAM 162 キャララタ・コード 73. 271 ンコンド 52 機長金 284 Wコマンド 52 機長金 62 Xコマンド 52 投口金 62 Xコマンド 52 ア・でサーラー 47 2-の機械を 20 20 20 の内機械を 20 20 20 あ行 マスキーコード 294 サーフェス 56 オースキーコード 294 サーフェス 56 オーン・アンドー 26 サーフェス 56 オーン・アンドー 26 サーフェス 56 オーン・アンドー 24 サーフェス 56 オーン・アンドー 70 86 カーリー・アドレス 249 ナーン・アンドー 70 86 カーリー・アドレス 249 カーカルケー 10 サールー・アドレス 249 カーリー・アドレス 249 カーカルケー 20 サールー・アドレス 21	のグラフィックス・ドット構成 159	画面コントロール 96
TM コマンド 51 キーフード 292 VRAM 162 キャラクタ・コード 73. 271 の移動 269 キャラクタ・コード 73. 271 Vコマンド 52 緑佐 264 VJマンド 52 緑佐 264 XOR命 128	- のメモリマップ28	間接アドレッシング240
VRAM 162 キャラクタ・コード 73、211 かの移動・269 Vコマンド・52 機成のか 62 XUコマンド・52 機成のか 62 XUコンド・52 機成のか 62 Xロコンド・52 アラフィンスモード・167 かかかかが 20 242 か行 スキュンド・29 242 サーフェス 56 インスキュンド・29 サーフェス 56 インスキュンド・50 ドロスキュンド・70、86 インストンド・70、86 インストンストントン・121	S コマンド 50	キーボード・データ117
- 一の移動・ 289	TM コマンド 51	₹-7- K 292
V コマンド 52 概性 284 Wコマンド 52 XOR命令 128 走アセンブラ 47 Xコマンド 52 グラフィッスモード 67 - で機用されるレツスタの記号・53 からよれ(オーバーフロー)・ 197 の内部構成・ 20 20 24 あ行 アスキュート 294 ヤーフェス 56 アスキュア・ 41 アセンブラ・ 41 アセンブラ・ 24, 35 アセンブリュ語・ 59 シータレット 70, 86 コーカかたち 64 フリド (市道)・ 195 コ世界・ 62 ソセンブテーブル・ 272 表現央的実性 66 センファ・ 272	VRAM 162	キャラクタ・コード 73, 271
N コマンド 52 接風を含 62 次 128 差 アセンブラ 47 7 7 7 7 2 2 2 2 2 2 3 4 2 3 4 3 4 3 4 3 4 3 4 3		キャラクタに色をつける102
XOR命令 128 ボアセンブラ 47 Xコマンド 52 グラフィックスモード 167 - で使用されるレジスクの記号・53 桁かよは(オーバーフロー) 197 Z-80A 20 20 20 21 64 あ行 アスキーコード 294 サーフェス 56 アスキースド 41 サーフェス 56 アセンブラ 24 35 博スタート・アドレス 249 アセンブリィ評 59 シーフレット 70,86 の出版 52 ジャンブラーブル 272 表現央的東軍 62 ジャンブラーブル 272 表現央的東軍 62 ジャンブラーブル 272	V コマンド 52	順度284
X コマンド 52 グラフィックスモード 167 - で使用されるレジスクの記号・53 Z-89A 20 20 242 カルラー 20 242 カイト 20 242	W コマンド 52	擬似命令62
- で使用されるレジスタの記号・53 おかねは(オーバーフロー) 197 Z-80A 20 20 247 2フィト 64 あ行 さ行 アスキーコード 294 サーフェス 56 アスキースケー 41 サイン・フラグ 27 アセンフリ、注: 59 シークレット・アドレス 249 アセンフリ、注: 59 シークレット 70,86 コガビラー 62 ジャンプラーブル 272 友現の約束等 62 ジャンプラーブル 272 麦現の約束等 60 ±シンスク・セット 271	XOR命令128	逆アセンブラ47
Z-80A 20 練算フラグ 27 の内部構成・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	X コマンド	グラフィックスモード167
あ行 20、242 コメント 64 あ行 さ行 た行 アスキーコード・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	で使用されるレジスタの記号…53	桁あふれ(オーバーフロー)197
あ行 アスキーコード 294 サーフェス 56 アスキース字 41 サイン・フラグ・27 アセンブラ 24, 35 ドステート・アドレス 249 アセンブリンボー 59 シークレット 70, 86 ー 7かた5 64 ションブラーフル 72 友現の約束等 62 ジャンプラーフル 272 麦現の約束等 60 ミンジステ・セット 272	Z-80A20	減算フラグ 27
アスキーフト 294 サーフェス 56 アスキース字 41 サイン・フラグ 27 アセンフラ 24 35 ドステート・アドレス 249 アセンブリ、(語 59 シータレット・ 70, 86 ー つかたた 64 ションド (附近り) 195 つ世界 62 ジャンプラーブル 272 差現央的東軍 60 主シンスタ・セット 271	の内部構成20, 242	コメント・・・・・・・・・・・・・・・・・64
アスキーフト 294 サーフェス 56 アスキース字 41 サイン・フラグ 27 アセンフラ 24 35 ドステート・アドレス 249 アセンブリ、(語 59 シータレット・ 70, 86 ー つかたた 64 ションド (附近り) 195 つ世界 62 ジャンプラーブル 272 差現央的東軍 60 主シンスタ・セット 271		
アスキー文字 41 サイシ・フラグ・・・・27 アセンプラー・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・		さ行
アセンフラ 24, 35 再スタート・アドレス 249 アセンフリ、計画 59 シークレット 70, 86 一のかたら 64 シフト(福港リ) 195 力世界 62 ジャンプラーブル 272 表現の約束率 60 主シンスタ・セット 211	アスキーコード294	
アセンブリス語・・・ 59 シータレット・・ 70, 86 - つかだち・・・ 64 シアト(前進り)・・ 195 の世界・・・ 62 ジャンプテーブル・ 272 表現の約束準・・ 60 主レジスタ・セット・ 21		
ーのかたち・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	アセンブラ 24, 35	
の世界		
表現の約束事		
アセンブルの手順		
	アセンブルの下順65	スタック・エリア168

スタック・ポインタ 242, 248
とメモリの関係244
スタック操作命令246
セクタ
セルフ・アセンブラ 65
ゼロ・フラグ
風性コード 75, 90
た行
中間言語
中間色284
直接アドレッシング240
テキストウィンドウ
テキスト側面とグラフィックス画面
の重ね合わせ 219
ディスプレイエリア 70
デジタルモード 284
トラック 56
ドット132
ドライブ132
tri
な行
ニーモニック・・・・・・・・・・・・・・・・・・・・・・・64
ニーモニック・・・・・ 64 →機械語対照表・・・・ 326
ニーモニック・・・・・ 64 →機械斯対照表・・・・ 326 は行
◆ 1
● 1
□ーゼニック 64 →・機械語対照表 326 は行 ハーフキャリー・フラグ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
ユーモニック 64 ・機模語対照表 326 は行 ハーフキャリ・・フラグ 27 ハーブペイト 17 ハンド・アセンブル 24 65 ンンドシェーク法 295
ニーモニック 64 ・機械部対照長 326 は行 ハーフキャリ・・フラグ・・・・27 ハーブバイト・・・17 ハンド・アセンブル・・28 バンドシェーク法 285 ハンドラ・・295
ユーモニック 64 ・機械請対照表 326 は行 ハーフキャリー・フラグ 27 ハーブバイト 177 ハンド・アセンブル 34.65 ハンドシェーグ法 295 ハンドシー 295 バックグランドのカラー化 103
ユーモニック 64 ・機模部対照長 326 は行 ハーフキャリー・フラグ 27 ハーブバイト 117 ハンド・アセンブル 24 65 ハンドシェーク法 295 バットグランドのカラー化 103 のカラーテクタ 97
ユーモニッタ 64 ・機械部対照表 326 は行 ハーフキャリ・・フラグ・27 ハーブイト 17 ハンド・アモンブル 24, 65 ハンドシェーク注 295 ハンドシ 295 バックグランドのカラー任 103 のカラーデータ 97 バックグランドカラ 290
ユーモニック 64 ・・機械語対照表 326 は行 ハーフキャリ・・フラグ 27 ハーブペイト 117 ハンド・アセンブル 28.65 ハンドシェーク法 295 ハンドシ 25 パックグランドのカラー化 103 ーのカラーデータ 97 パックグランドカラー 290 パンクメモリ 255
ユーモニック 64 ・・機械部列照表 326 は行 ハーフキャリ・・フラグ・・・・27 ハーブペイト 117 ハンド・アセンブル 24 65 ハンドシェーク法 295 バックグランドのカラー化・・・103 ーのカラーデータ 97 バックグランドカラー 290 バックグランドカラー 290 バックグランドカラー 27 ベックグランドカラー 28 ベックグランドカラー 28 ベックグランドカラー 28 ベックグランドカラー 28 ス・ス・ス・ス・ス・ス・ス・ス・ス・ス・ス・ス・ス・ス・ス・ス・ス・ス・ス・
ユーモニック 64 ・・機械語対照表 326 は行 ハーフキャリー・フラグ 27 ハーアパイト 17 ハンド・アセンブル 24・65 ハンドシューク法 295 ハンドシー 395 インタグランドのカラーデータ 97 パックグランドカラー 290 パンクグランドカラー 290 パンクグモリ 25・102・254 バンクリー 25・102・254
・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
ユーモニック 64
・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
ユーモニック 64

プリンク 70, 8
ブレイク・ポイント 44, 5
ブロック転送命令97, 27
ブレーン16
と色の関係16
プログラム・カウンタ 45、24
プログラムリストの見方 8
ペアレジスタ24
の退避24
ポインタ 20
棚助レジスタ・セット2
暴走25
ま行
マシン語プログラムエリア3
マシン語モニタの起動 3
の特長3
マシン語命令の長さ2
未使用命令 271, 27
メモリモードの切り替え25
メモリ内容の変更 4
モニタに入る27
に戻る24
ら行
ラベル・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
リバース 70, 77, 8
リンクポインター・・・・・・26
レジスタ退避命令24
論理演算の特殊な計算13
わ行
ワークエリア・・・・・・・・・26
401 i) 13, 21 94

前田 光男(まだ みつお) 昭和27年6月10日生札、東京連科大学理 学部中、現在、東京都江東区立や町中学 校敷施、同板で科学部の期間をし、生徒に バーンナルコンピュータを指導・、著書 に「マシン語でゲームを作る本」「mkII FAN BOOKS 2]マシン語入門」、(技術計論 社)

PC-8801mk]] SR SR SUPER BOOKS② やさしいマシン語

昭和60年 8 月20日 初版 第 1 断発行

署 者 前田 光男 発行者 片岡 麗 発行所 株式会社 技術評論社 東京都千代田区平河町1-4-12 電話 03(262)9351 営業部 03(273)8315 編集部

印 刷 加藤文明社

定価はカバーに表示してあります。

本書の一部または全部を著作権法の定める 範囲を超え、無断で複写、複製、転載、テ 一ブ化、ファイルに落すことを禁じます。 (C1985 前田光男

ISBN4-87408-279-3 C3055



SUPER BOOKS

初級者から中級者までのマシン語

2.
モノクロ画面からカラー画面への招待

3.
キーボードを操作する

4.
グラフィックスの世界

スムーズ・スクロール

マシン語アラカルト